TEXAS INSTRUMENTS

# MEMORY

# GENERAL DESCRIPTION

# EXPLORER™ MEMORY GENERAL DESCRIPTION

# MANUAL REVISION HISTORY

Explorer™ Memory General Description (2533592-0001)

Original Issue .....August 1985



The computers, as well as the programs that TI has created to use with them, are tools that can help people better manage the information used in their business; but tools—including TI computers—cannot replace sound judgment nor make the manager's business decisions.

Consequently, TI cannot warrant that its systems are suitable for any specific customer application. The manager must rely on judgment of what is best for his or her business.

Printed in U.S.A.

# THE EXPLORER™ SYSTEM HARDWARE MANUALS

**System Level Hardware Publications**

Explorer Unpacking and Inventory ..... 2243216-0001
Explorer 7-Slot System Installation ..... 2243140-0001
Explorer System Field Maintenance ..... 2243141-0001

**System Enclosure Hardware Publications**

Explorer 7-Slot System Enclosure General Description ..... 2243143-0001
Explorer 2-Megabyte Memory General Description ..... 2243147-0001
Explorer Memory General Description ..... 2533592-0001
Explorer Processor General Description ..... 2243144-0001
Explorer Display Unit General Description ..... 2243151-0001
Explorer System Interface General Description ..... 2243145-0001

**Mass Storage Hardware Publications**

Explorer Mass Storage Enclosure General Description ..... 2243148-0001
Explorer Winchester Disk Formatter (ADAPTEC) Supplement to Explorer Mass Storage Enclosure General Description ..... 2243149-0001
Explorer Winchester Disk Drive (Maxtor) Supplement to Explorer Mass Storage Enclosure General Description ..... 2243150-0001
Explorer Cartridge Tape Drive (Cipher) Supplement to Explorer Mass Storage Enclosure General Description ..... 2243166-0001
Explorer Cable Interconnect Board (2236120-0001) Supplement to Explorer Mass Storage Enclosure General Description ..... 2243177-0001
Explorer NuBus™ Peripheral Interface General Description (NUPI board) ..... 2243146-0001

**Mass Storage Hardware Vendor Publications**

Series 540 Cartridge Tape Drive Product Description, Cipher Data Products, Inc., Bulletin Number 01-311-0284-1K (¼-inch tape drive) .. 2249997-0001
MT01 Tape Controller Technical Manual, Emulex Corporation, part number MT0151001 (formatter for the ¼-inch tape drive) ..... 2243182-0001
XT-1000 Service Manual, 5¼-inch Fixed Disk Drive, Maxtor Corporation, part number 20005 (5¼-inch Winchester disk drive, 112 megabytes) ..... 2249999-0001
ACB-5500 Winchester Disk Controller User's Manual, Adaptec, Inc., (formatter for the 5¼-inch Winchester disk drive) ..... 2249933-0001

**Optional Equipment Hardware Publications**

Explorer NuBus Ethernet® Controller General Description ..... 2243161-0001
Model 855 Printer Operator's Manual ..... 2225911-0001
Model 855 Printer Technical Reference Manual ..... 2232822-0001
Model 855/856 Printer Maintenance Manual ..... 2225914-0001

# THE EXPLORER™ SYSTEM SOFTWARE MANUALS

**Mastering the Explorer Environment**

| | |
|---|---|
| Explorer Technical Summary | 2243189-0001 |
| Explorer Operations Guide | 2243190-0001 |
| Explorer Zmacs Editor Tutorial | 2243191-0001 |
| Explorer Glossary | 2243134-0001 |
| Explorer Communications User's Guide | 2243206-0001 |
| Explorer Diagnostics | 2533554-0001 |
| Explorer Master Index to Software Manuals | 2243198-0001 |
| Explorer System Software Installation | 2243205-0001 |

**Programming With the Explorer**

| | |
|---|---|
| Explorer Programming Primer | 2243199-0001 |
| Common LISP, The Language, by Guy L. Steele, Jr. | 2537175-0001 |
| Explorer Lisp Reference | 2243201-0001 |
| Explorer Zmacs Editor Reference | 2243192-0001 |
| Explorer Programming Concepts and Tools | 2243130-0001 |
| Explorer Window System Reference | 2243200-0001 |
| Explorer Command Interface Toolkit User's Guide | 2243197-0001 |

**Explorer Toolkits**

| | |
|---|---|
| Explorer Natural Language Menu System User's Guide | 2243202-0001 |
| Explorer Relational Table Management System User's Guide | 2243203-0001 |
| Explorer Graphics Toolkit User's Guide | 2243195-0001 |
| Explorer Grasper User's Guide | 2243135-0001 |
| Explorer Prolog Toolkit User's Guide | 2243204-0001 |
| Programming in Prolog, by Clocksin and Mellish | 2249985-0001 |
| Explorer Color Graphics User's Guide, Support for the Raster Technologies Model One | 2537157-0001 |
| Explorer TCP/IP User's Guide | 2537150-0001 |

**System Software Internals**

| | |
|---|---|
| Explorer System Software Design Notes | 2243208-0001 |

# ABOUT THIS MANUAL

## Introduction

This document describes the memory board used in the Texas Instruments Explorer computer system. The intended audience for this document includes original equipment manufacturers (OEMs), system designers, field maintenance personnel, TI customer representatives, and people in similar groups.

## Contents of This Manual

This document consists of four sections. A brief description of each section is as follows:

**Section 1:** Introduction — Provides general information on the Explorer memory board.

**Section 2:** Installation — Provides unpacking, installation, and removal procedures for the memory board.

**Section 3:** Operating Instructions — Provides self-test operating instructions for the memory board.

**Section 4:** System Design and Programming Data — Provides a block diagram description of the memory board, a general description of the local bus and NuBus data transfers, and programming information.

# CONTENTS

# 1 INTRODUCTION

**Highlights of This Section**

- General information
- Features
- Specifications
- Performance

## General

**1.1** This section provides general information on the Explorer memory board shown in Figure 1-1. The components of an Explorer computer system are shown in Figure 1-2. Figure 1-3 shows the system enclosure in more detail.

The memory board is a standard three-high Eurocard design with three standard DIN connectors referred to as P1, P2, and P3. The expression Eurocard indicates a European mechanical-design standard for circuit boards. The number of connectors a board has space for indicates the height of the board. For example, a one-high Eurocard has space for one connector and is 100 millimeters (3.9 inches) high and 280 millimeters (11 inches) in depth. A three-high Eurocard has space for three connectors and is 366.7 millimeters (14.4 inches) high. All Eurocards have the same depth. In addition, the memory board has special ejector/injector mechanisms that provide easy insertion and removal of the board from the mating connectors on the backplane of the system enclosure.

The memory board has a capacity of two megabytes when using 64K-bit (where K equals 1024) dynamic random-access memory (DRAM) chips, and a capacity of eight megabytes when equipped with 256K-bit DRAM chips. Explorer memory boards are also available in one-megabyte and four-megabyte capacities.

**Figure 1-1 Explorer Memory Board**

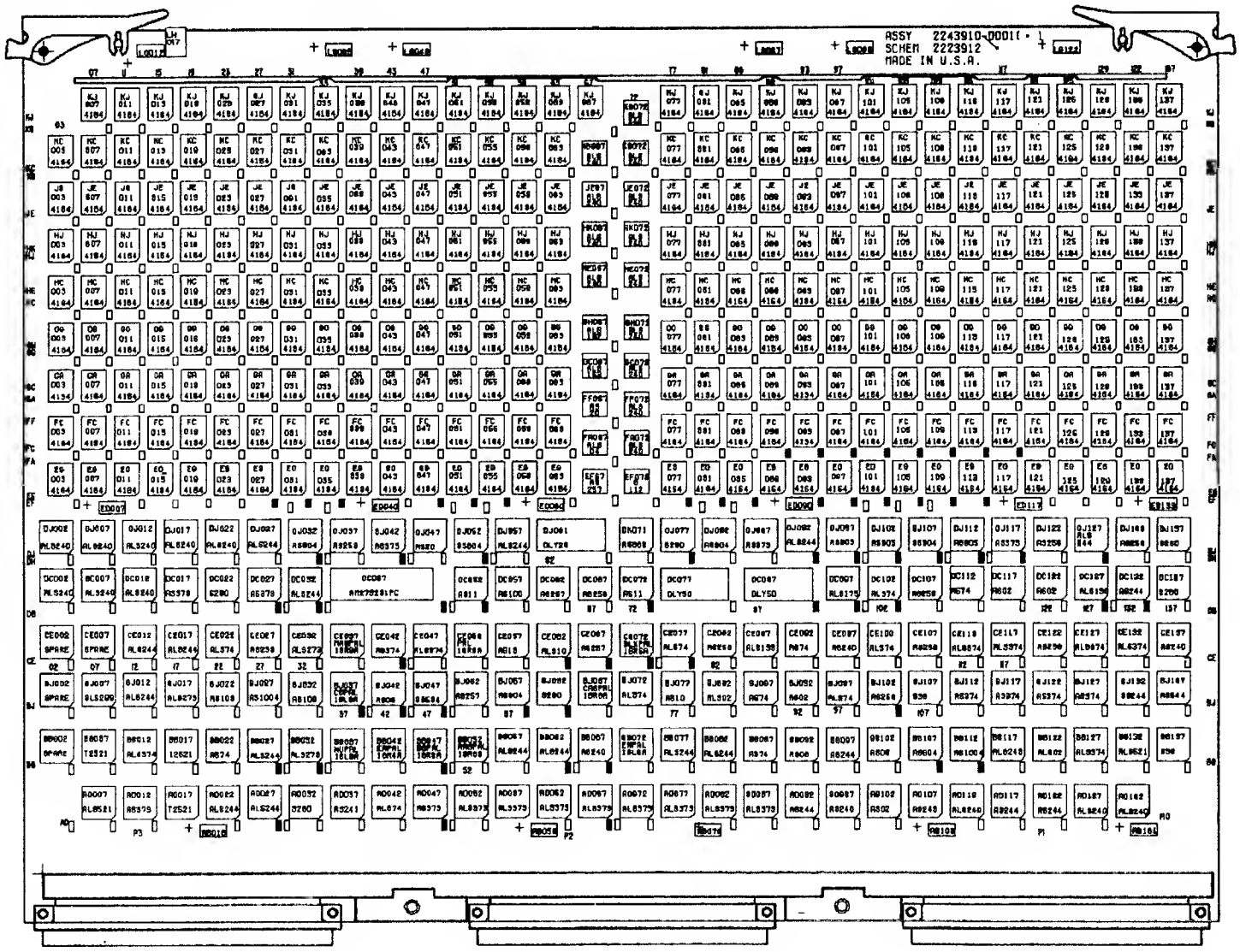


**Figure 1-2 Example of the Explorer System Components**

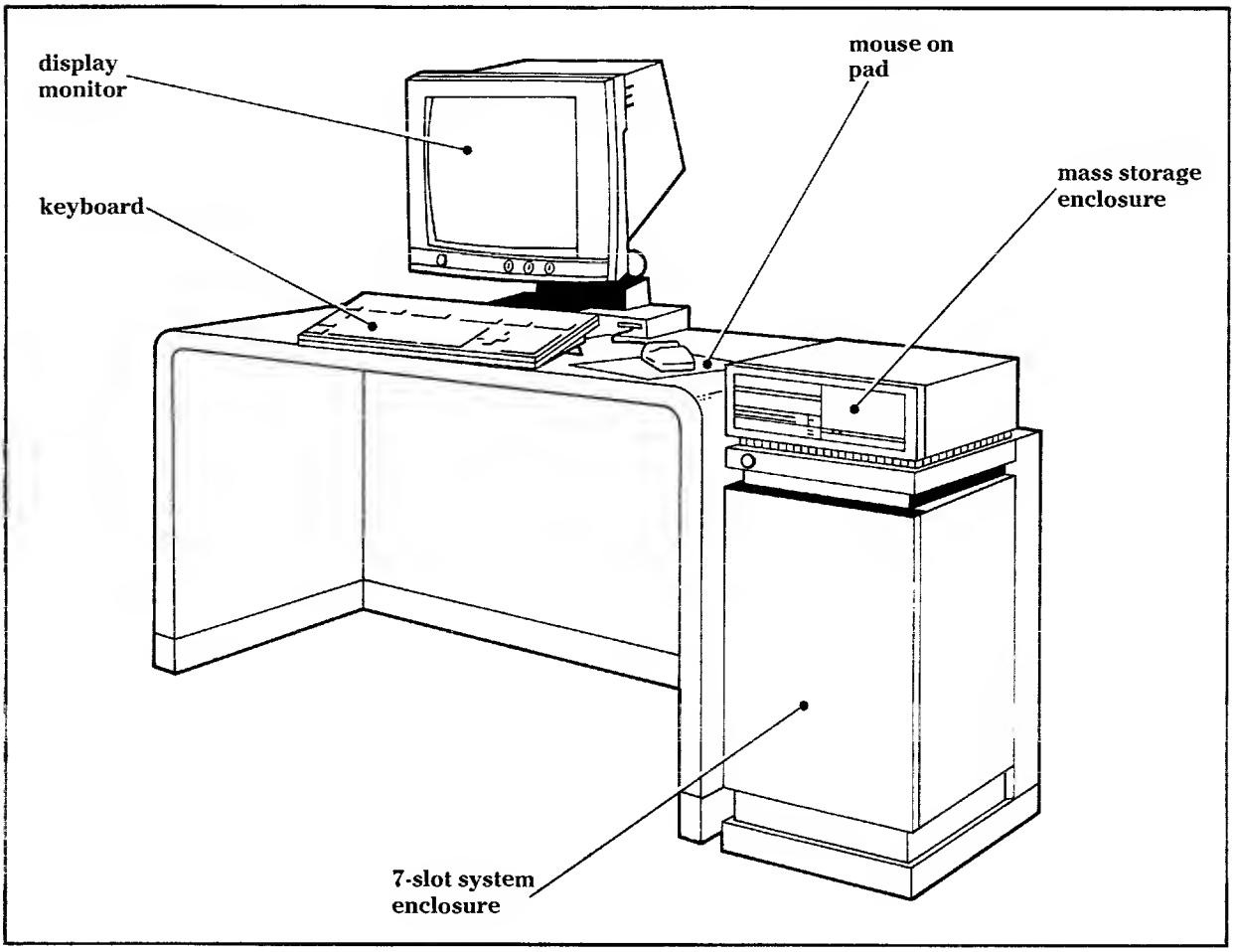
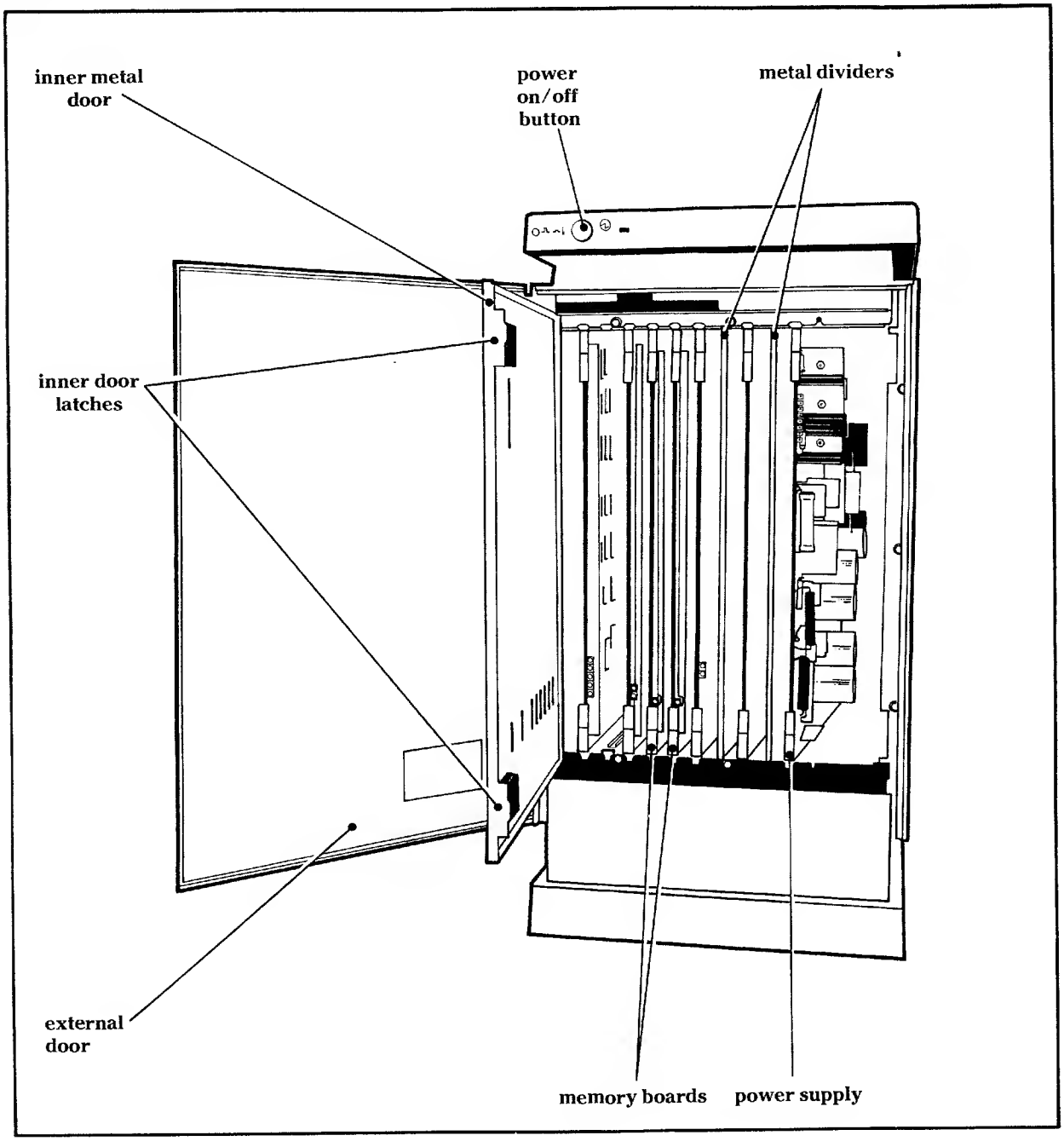


**Figure 1-3 Explorer System Enclosure**



The memory board communicates with other components in the system over the local bus and the NuBus communication paths. The NuBus is a high-speed synchronous bus that multiplexes 32-bit data words with 32-bit address codes. It has a simple communications protocol that allows the circuit boards on the bus to be master or slave devices over the bus. The local bus is also a high-speed synchronous bus, but it provides separate paths for the 32-bit data words and 32-bit address codes. The memory board contains the circuits that allow the processor on the local bus to communicate with the NuBus as a master or slave.

Although some of the circuit boards can be inserted in any slot in the system enclosure, there are certain boards that should always be located in particular slots. A typical board configuration is as follows:

- Slot 0 — NuBus Ethernet controller board (if Ethernet is present)
- Slot 1 — Optional slot
- Slot 2 — NuBus peripheral interface (NUPI) board
- Slot 3 — Memory board
- Slot 4 — Memory board
- Slot 5 — System interface board (SIB)
- Slot 6 — Processor board
- Slot 7 — Power supply

The assigned slots are:

- Slot 6 is reserved for the processor board.
- Slot 5 should contain the system interface board to keep it close to the processor.
- Slot 4 must contain a memory board, which becomes the master memory board when two or more are present.

Slots 0, 1, 2, and 3 are nonrestrictive; that is, any of the boards not assigned to slots 4, 5, and 6 can be located in any of the remaining slots.

The slots are identified by a 4-bit identification (ID) code that represents hexadecimal numbers 0 through F on the backplane. This ID code becomes part of the board address configuration when the board is inserted into a slot. The board slot address configuration is:

>FSXXXXXX

where:

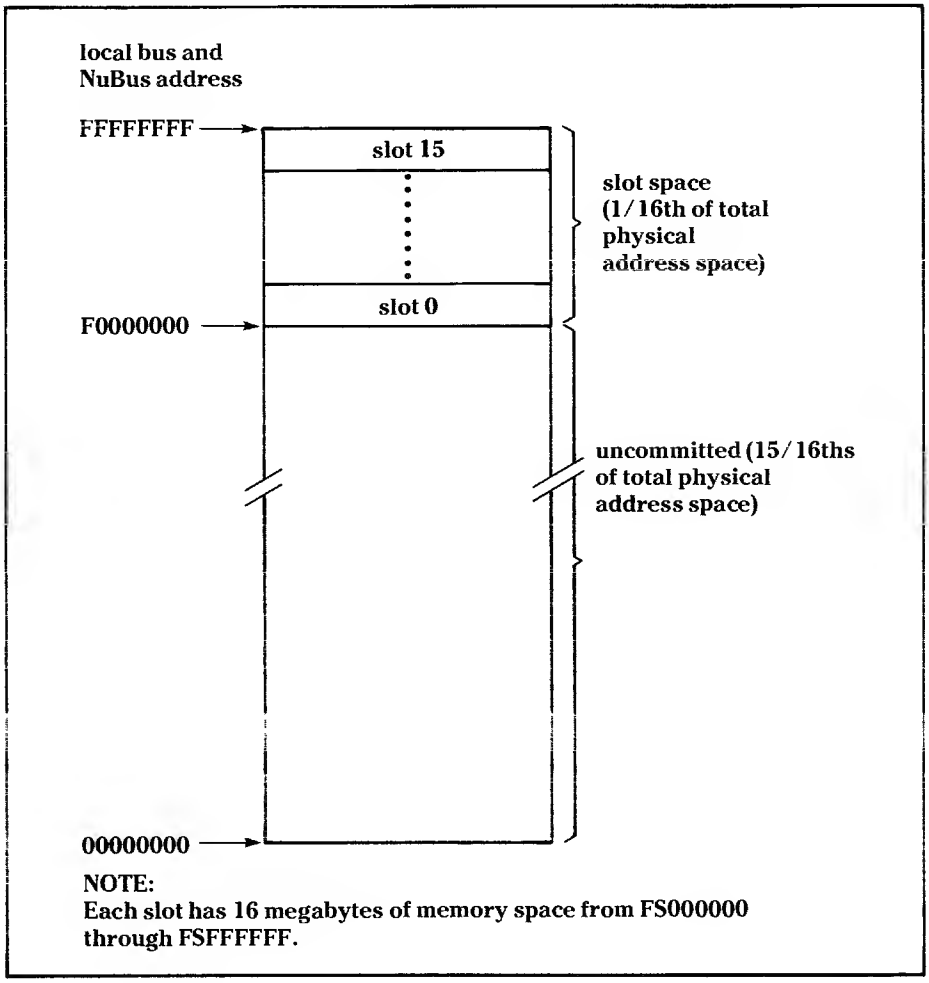
F is a hexadecimal value.
S is the slot ID.
X is a variable, 0 through F.

A configuration read-only memory (ROM) is provided on each circuit board. This ROM contains a unique name to identify the specific board of which it is a part. This name is addressed along with the slot ID during power-up so that the processor knows the type of circuit board in each slot during all succeeding address operations on the NuBus and the local bus.

Each slot has 16 megabytes of address space from >FS000000 through >FSFFFFFF. These address spaces are distributed to the 16 possible slots on the address space map shown in Figure 1-4. The 16 slots allow a total address capacity of 256 megabytes. This is only 1/16th of the total address capacity of the full 32-bit NuBus and local bus, which is 4 gigabytes. The unused address space is uncommitted and allocated as required.

The local bus uses only four slots on the Explorer system enclosure, which limits the address capacity to 64 megabytes. The local bus has the capacity to address four gigabytes when using the full 32-bit address and the base register feature that allows for memory address mapping.

Figure 1-4 **Local Bus and NuBus Address Space Map**




## Features

**1.2** The following briefly describes the memory board features:

- Each memory board that is fully equipped with 64K-bit memory chips (where K equals 1024) has a capacity of 2 megabytes.
- The memory board can use 256K-bit memory chips with only factory jumper modifications.
- Each word in memory consists of 32 data bits (4 bytes). Four parity bits are generated for each word. These parity bits are stored in separate memory chips for use in diagnostics.
- Data is sent over the NuBus and local bus in 32-bit words, 16-bit halfwords, or 8-bit bytes.

- The memory board provides a communication interface to the NuBus and the local bus.
- The memory space can be accessed from either the NuBus or the local bus.
- The memory board is a slave to the NuBus except when the local bus processor has control of the memory board. The memory board can then become the NuBus master by arbitrating for NuBus control.
- The memory board is always a slave to the local bus when the local bus is provided.
- When two memory boards are used in a system with a NuBus and a local bus, one board can access the NuBus at the same time the other board is accessing the local bus.
- The slot address of the memory board is determined by four ID bits on the NuBus that are connected to ground at each connector to make the ID code for that slot.
- The memory board contains a base register that allows data from the NuBus or the local bus to change the address range of the memory board.
- Memory data block transfers are supported when the memory board is functioning as a NuBus slave. Block transfers are not supported on the local bus.
- The memory board provides a locking circuit that can lock out the NuBus or the local bus so that either bus can perform continuous read/write cycles.
- The memory board uses the clock signals from the NuBus and local bus. The NuBus clock must be available at all times. Most circuitry is synchronized with a board clock which is driven by the clock for the bus which is currently accessing the memory board.

## Specifications

**1.3** Table 1-1 lists the general specifications for the Explorer memory board.

**Table 1-1** **Memory Board Specifications**

| *Item* | *Specifications* |
|---|---|
| Power usage: | |
| Voltage | 5 Vdc |
| Typical current | 9.5 A (with 2 megabytes)<br>10.1 A (with 8 megabytes) |
| Ambient temperature: | |
| Operating | 10 to 35° C (50 to 95° F) |
| Nonoperating | −40 to 65° C (−40 to 149° F) |
| Relative humidity: | |
| Operating | 15 to 80% noncondensing |
| Nonoperating | 5 to 95% noncondensing |
| Dimensions: | |
| Width | 366.7 mm (14.437 in) |
| Depth | 280 mm (11.024 in) |
| Thickness (with components) | 18.11 mm (0.714 in) maximum |
| Memory capacity (maximum) | 2 megabytes (with 64K-bit chips)<br>8 megabytes (with 256K-bit chips) |
| Logic states: | |
| Unasserted (high false) | >2.0 V at the receiver |
| Asserted (low true) | <0.8 V at the receiver |
| Local bus clock frequency | 7 to 10 MHz (processor determined) |
| NuBus clock frequency | 10 MHz |
| Electromagnetic emission | Complies with FCC level B,<br>EMI/RFI office emission requirements |

## Performance

**1.4** The performance characteristics of the Explorer memory board are explained in the following paragraphs.

### Board Clock Frequency

**1.4.1** The board clock can be driven from the local bus at a frequency from 7 to 10 megahertz, or it can be driven from the NuBus at a frequency of 10 megahertz.

### Amount of Bus Arbitration

**1.4.2** The memory board selects the NuBus after initial power-up or NuBus board reset. After that the last bus to access the memory board remains selected. Time spent in selecting and enabling the requesting bus, if necessary, is approximately equal to one memory access time (four or five clock periods).

Simultaneous requests for memory access from the local bus and the NuBus result in the selection of the competing bus after each access, until only one bus is left with a request. This allows equal access to either bus; however, it results in bus selection time between accesses.

The information in the following two paragraphs explains the access times, assuming that the memory board is already synchronized to the requesting bus.

### Memory Access From the Local Bus

**1.4.3** In memory read operations, data is valid 250 nanoseconds after the falling edge of the local bus clock. BERR is valid 284 nanoseconds after the same falling edge.

In memory write operations, MEMREQ– must be held asserted until two clock periods after the falling edge of the local bus clock. Data and address line values are irrelevant 50 nanoseconds after the first falling edge of the local bus clock.

### Memory Access From the NuBus

**1.4.4** In single memory read operations, data and control lines are valid on the fifth sample edge after the START– signal.

In single memory write operations, data and control lines are valid on the fourth sample edge after the START– signal.

In block memory read operations, data and control lines are valid on the fifth sample edge and on every third sample edge after that until the completion of a transfer.

In block memory write operations, data and control lines are valid on the fourth sample edge and on every other sample edge after that until the completion of a transfer.

**Tables**

# INSTALLATION 2

**Highlights of This Section**

- Introduction
- Unpacking procedure
- Installation and removal procedure

## Introduction

**2.1** This section provides unpacking, installation, and removal instructions for the Explorer memory board.

**CAUTION: The memory board contains static-sensitive electronic components. To prevent damage to these components, make sure that you are properly grounded before handling the memory board.**

**The recommended grounding method is to use a static-control system composed of a static-control floor or table mat and a static-control wrist strap (these are commercially available). If you do not have a static-control system, you can discharge any static charge by touching a properly grounded object prior to handling the memory board. Then, as an additional safety measure, put the memory board on a grounded work surface after removing it from the system enclosure or its static-protective package.**

**Before storing or transporting the memory board, return it to its static-protective package or the system enclosure.**

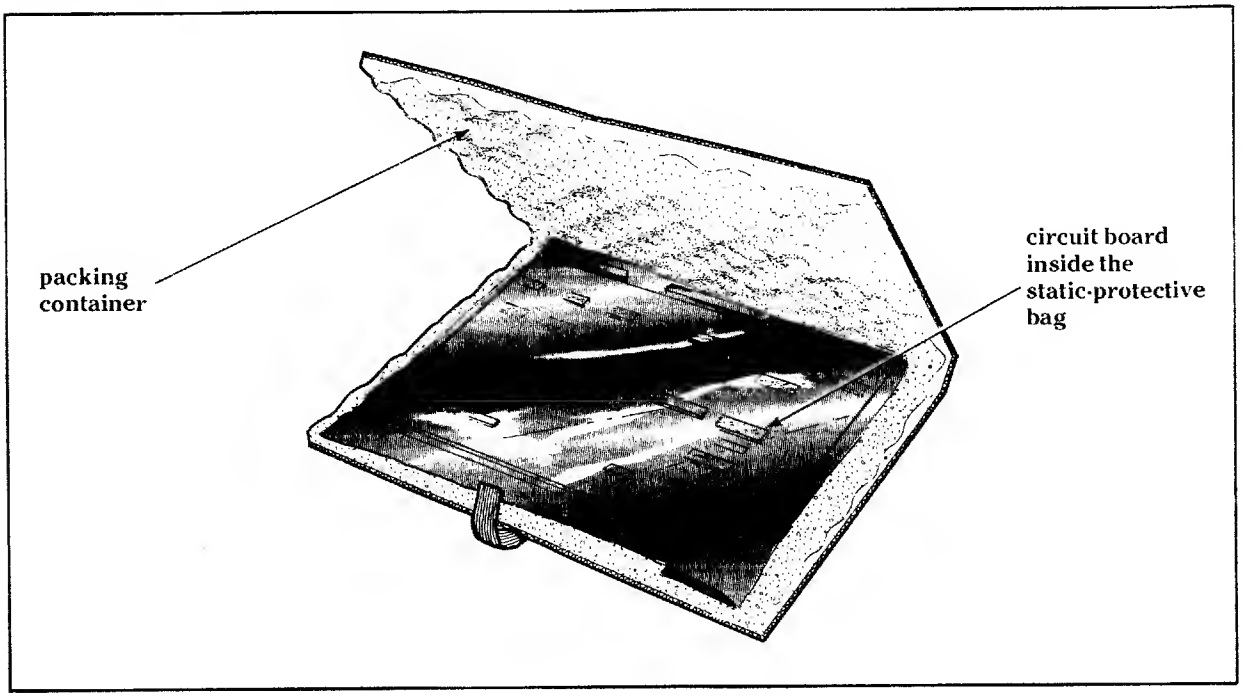
## Unpacking Procedure

**2.2** Unpack and visually check the memory board as follows:

1. Check for any documents fastened to the exterior of the packing container. If any documents are found, read and follow the instructions that are given.

2. Open the packing container (Figure 2-1) and carefully remove the packing material; then remove the memory board with the static-protective bag in place.

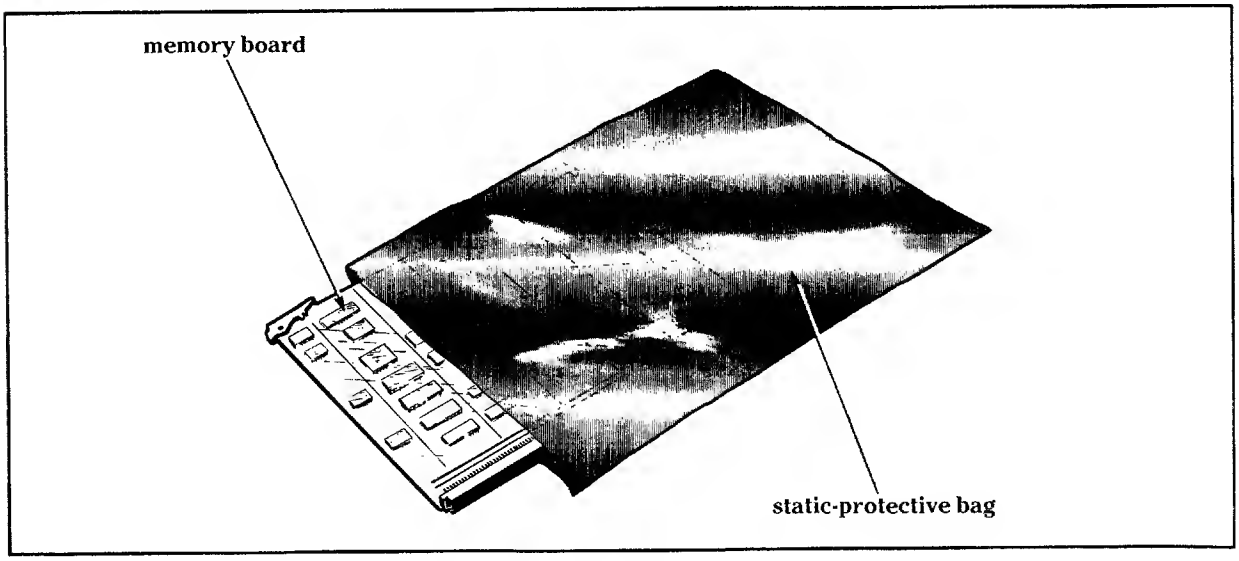
**Figure 2-1 Memory Board in Packing Container**




3. Remove the static-protective bag (Figure 2-2) from the memory board. Be sure to follow the static-caution recommendations when handling the memory board without the static-protective bag.

4. Check the memory board for scratches, broken parts, marred finish, or any other damage that may have occurred during shipment. Follow your local procedures to report any damage.

**Figure 2-2 Removal of Static-Protective Bag**



## Installation and Removal Procedure

**2.3** Install or remove a memory board as follows:

---

**NOTE:** There are no jumpers or switches to set or check prior to installing the memory board.

---

1. Set the power switch on the system enclosure to the off position. The power on/off switch is a pushbutton at the upper left-hand corner of the front of the system enclosure. The in position is on, the out position is off.

2. Open the front door of the system enclosure.

3. Release the latches on the internal metal door, and open this door.

4. Slide the memory board into the proper slot and into the connectors on the backplane. Lock the board into the slot connectors on the backplane using the two injector/ejectors on the ends of the board. The injector/ejectors are also used to remove the board from the backplane connectors.

---

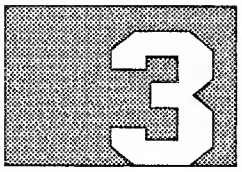
**NOTE:** To remove the board, follow steps 1 through 4 except in step 4 slide the board out of the slot instead of into the slot.

---

5. If you are ready to switch on the system, leave the outer door ajar so you can see the fault indicator light emitting diodes (LEDs) near the edges of the circuit boards. For power-up and self-test procedures, please refer to Section 3.

6. After the self-tests are completed and you are ready for routine operation, close the front door of the system enclosure.

# OPERATING INSTRUCTIONS



**Highlights of This Section**

This section describes system start-up procedures, the self-tests that occur automatically, and the meaning of red fault LEDs going off or staying on.

## Introduction

**3.1** This section describes the memory board self-tests that occur automatically when power is first applied to the system.

## Self-Test Procedures

**3.2** When power is first applied to the Explorer system, a number of self-tests occur automatically. The results of these self-tests are shown on the video display as they are completed. The automatic self-test does not test the entire memory board, but an extended self-test is available which does. This is mentioned at the appropriate point in the self-test procedure that follows:

---

**WARNING: Except during the self-test period when the front door is left open to observe the fault LEDs, do not operate the computer system with doors and/or panels of the system enclosure open. Under normal conditions, interlocks prevent power from being applied when these panels are not in place and the inner metal door is open. Do not bypass or otherwise tamper with the interlocks. Potentially lethal voltages are exposed if this warning is not observed.**

---

1. Set the power switch to the on (in) position. The power on/off switch is a pushbutton near the upper left-hand corner of the front of the system enclosure.

2. Check the red fault LEDs on the circuit boards in the system enclosure while the self-tests are running. The fault LEDs are located at the lower front edges of the boards, as shown in Figure 3-1.

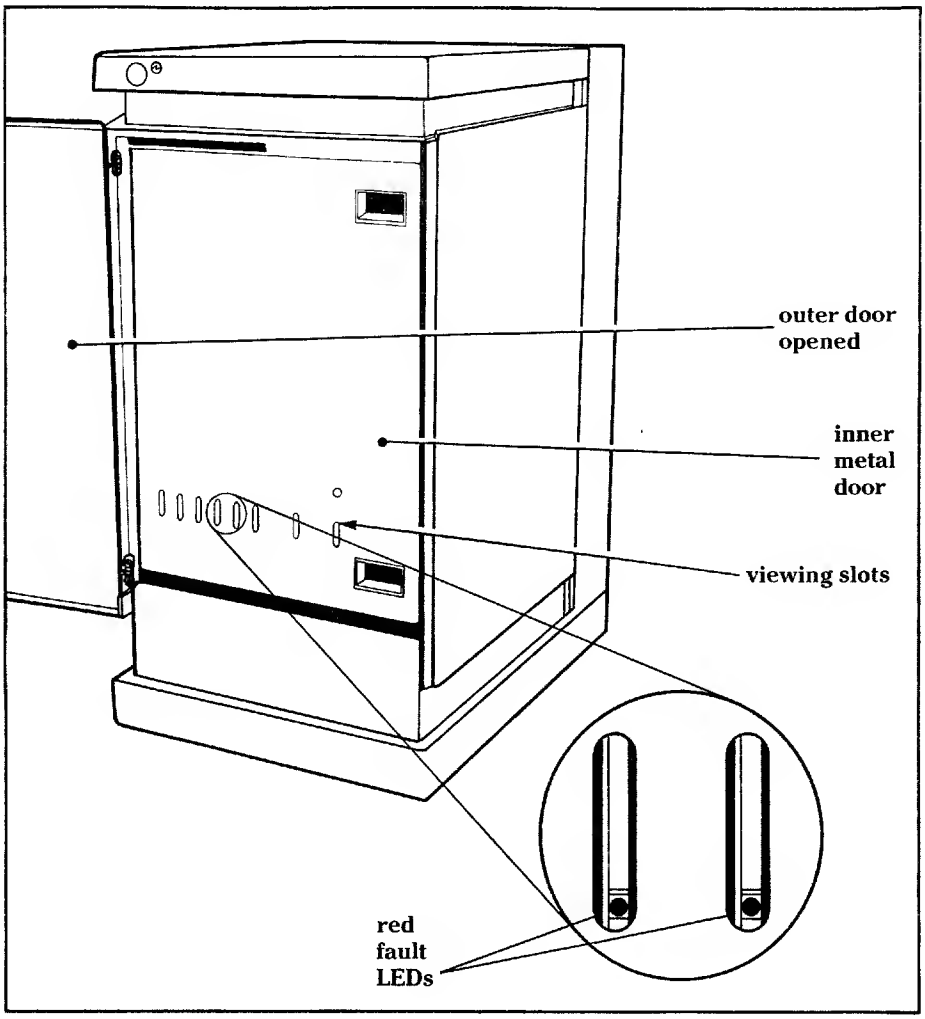
3. The self-test microcode on the processor board runs first. If the red fault LED on the processor board goes out, the processor board is good. The self-tests on the processor board then run the self-test microcode on the memory board.

4. If the red fault LED on a memory board goes out, the memory board is good.

5. If the red fault LED on a memory board remains on after the self-test, the board is probably faulty.

6. The automatic self-test does not test the entire memory board. Before proceeding further, you may want to press the E key to run the extended mode self-tests.

**Figure 3-1** **Fault LED Locations**

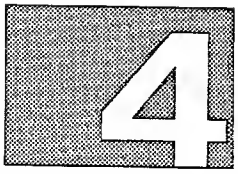



7. If the memory board in slot 4 fails, it can be temporarily replaced with the board from slot 3. (There must be a good memory board in slot 4 for the system to operate.) If the memory board in slot 3 fails, it can be laid aside until a replacement is available. The system can be used, but with a reduced amount of memory, as long as a good memory board is located in slot 4.

8. If the red fault LEDs on the processor board and the memory board remain on, then a fault may exist on the processor board, memory board, or other system component.

9. Refer to the installation and removal procedure in Section 2 to replace the faulty board.

# SYSTEM DESIGN AND PROGRAMMING DATA



**Highlights of This Section**

- Introduction
- Block diagram description
- Data transfer operation
- Programming information

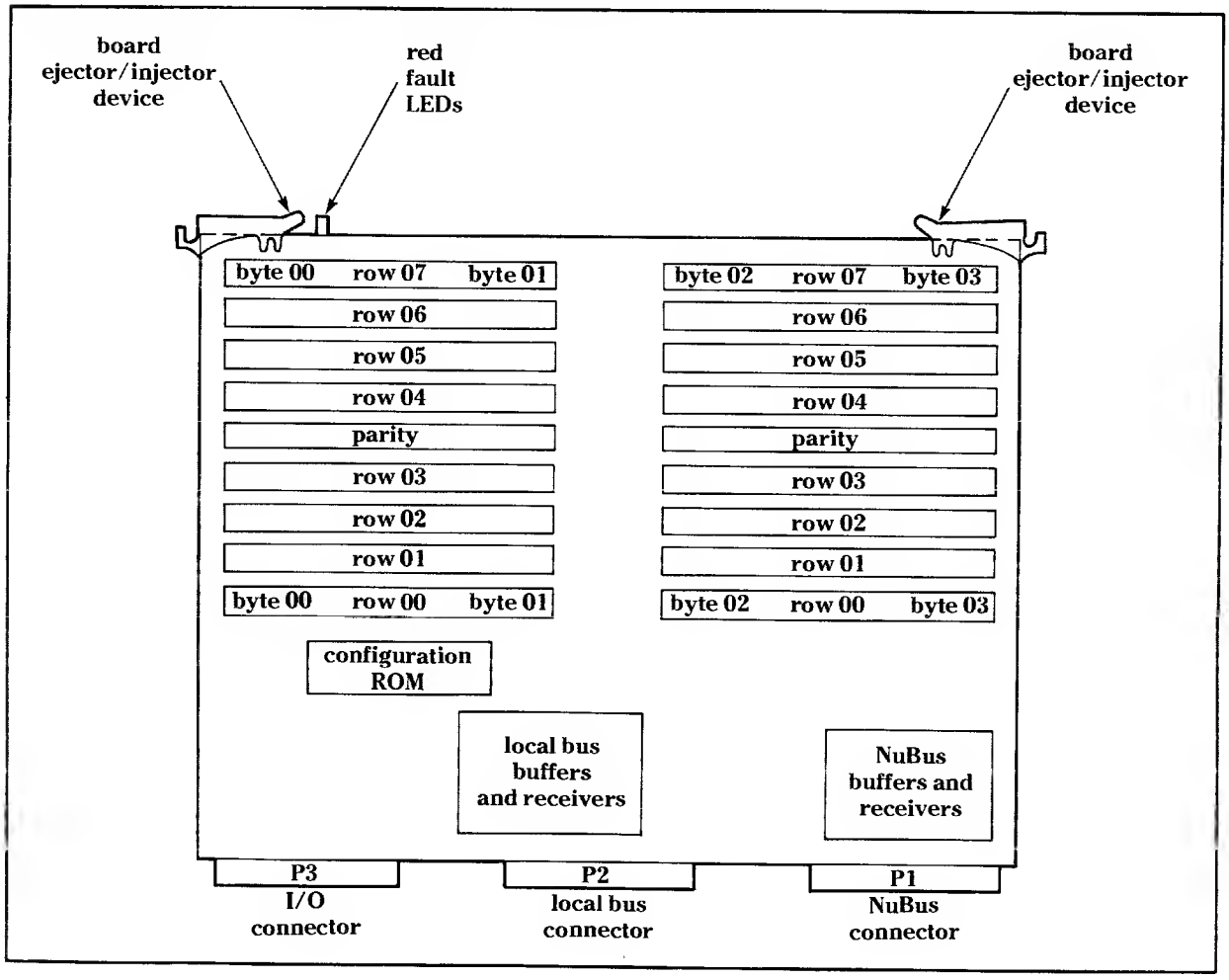
## Introduction

**4.1** This section provides general operating and design information for the Explorer memory board under the following topics:

- Block diagram description
- Data transfer operation
- Programming information

Figure 4-1 shows the physical organization of the components on the memory board. There are eight rows of memory chips defined as rows 00 through 07, with one row provided for parity. Each memory and parity row contains 32 memory chips. The memory rows are divided into four groups of 8 chips. The four groups are then defined as bytes 00 through 03, indicating the 4 bytes of a 32-bit word. Using 64K-bit chips, there are 256K bytes per row of chips, providing 2 megabytes of memory in eight rows. Using 256K-bit chips, there is 1 megabyte per row of chips, providing 8

**Figure 4-1 Memory Board Physical Organization**



megabytes of memory in eight rows. The boards can also be half equipped with chips, which allow memory capacities of 1 megabyte and 4 megabytes.

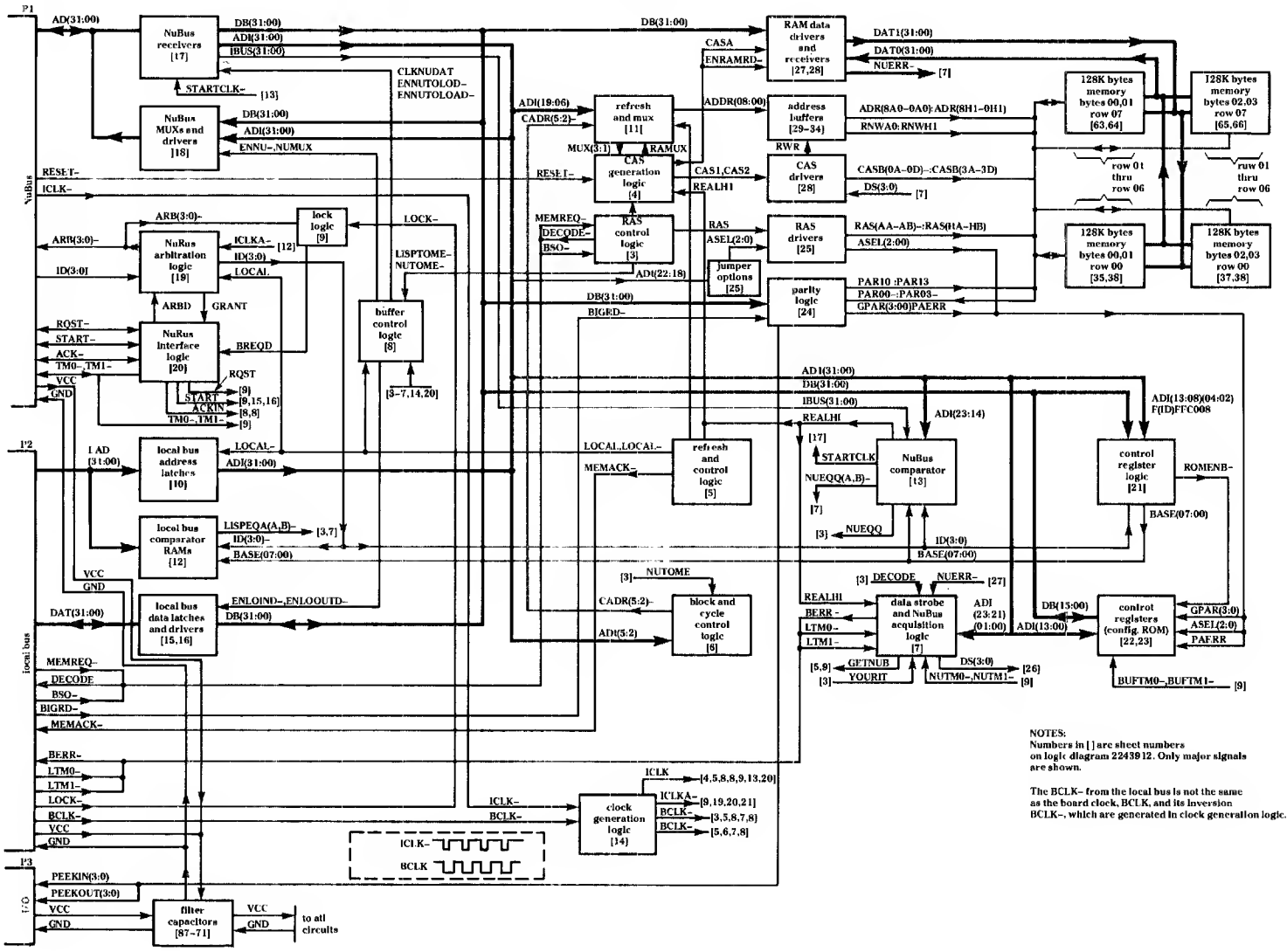
The other logic on the board includes drivers, control logic, a configuration read-only memory (ROM), and local bus and NuBus buffers and drivers. The memory board also has one self-test fault LED, three connectors defined as P1, P2, and P3, and two board ejector/injector devices.

## Block Diagram Description

**4.2** Use the memory board block diagram (Figure 4-2) as a guide in the analysis of the following memory board operations:

- Module slot addressing
- Base address register
- Organization of words, halfwords, and bytes
- Data transfer mode signals
- NuBus status code signals
- NuBus block sizes and starting addresses
- Clock generation
- Bus arbitration logic
- NuBus arbitration
- Control logic
- Local bus and NuBus locking
- Jumper options
- Memory address development
- Memory refresh cycle
- Local bus and NuBus signal definitions
- Connector pin/signal assignments

## Figure 4-2 Memory Board Block Diagram



**Module Slot Addressing**

**4.2.1** The memory board contains the NuBus master interface for the processor on the local bus. The NuBus master interface is enabled by the BS0– signal that is connected only to slot 4 on the local bus backplane in the system enclosure. The memory board in slot 4 is then the NuBus master for the processor on the local bus when BS0– is asserted. When there are two memory boards in a chassis with a NuBus and a local bus, one board can service the NuBus at the same time the other board is servicing the local bus.

The slot address of the memory board is determined by the four ID(3:0)– bits on the backplane that are related to the slots on the NuBus as shown in Table 4-1. Asserted bits in the hexadecimal ID(3:0)– code are connected to ground. For example, hexadecimal code F has all four bits connected to ground, and none of the four bits in hexadecimal code 0 are connected to ground.

**Table 4-1 NuBus Slot Number/ID Code Assignments**

| *Hexadecimal ID 10(3:0) – Code* | *Slot Number* |
|---|---|
| 0 | 0 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |
| 6 | 6 |
| 7 | 7 |
| 8 | 8 |
| 9 | 9 |
| A | 10 |
| B | 11 |
| C | 12 |
| D | 13 |
| E | 14 |
| F | 15 |

**NOTE:**

Slots are numerically listed from left to right as viewed from the front of the system enclosure. Although the Explorer system enclosure uses only seven addressable slots, the slot IDs can accommodate up to 16 addressable slots.

A reference to any address >FSXXXXXX will select the memory board. Addresses starting at >FSFFFFFF and decreasing through >FSFFC000 make up the control area and contain the configuration ROM and the on-board registers. Memory data addresses for the different sizes of memory boards are as follows:

| *Memory Size* | *Memory Data Addresses* |
|---|---|
| 1 megabyte | >FS000000 through >FS0FFFFF |
| 2 megabytes | >FS000000 through >FS1FFFFF |
| 4 megabytes | >FS000000 through >FS3FFFFF |
| 8 megabytes | >FS000000 through >FS7FFFFF |

**Base Address Register**

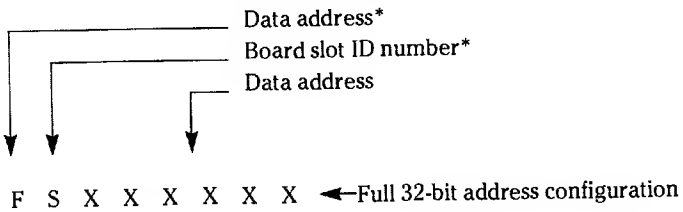
**4.2.2** The control register logic on the memory board contains a physical base register that has a BASE(07:00) code output. The function of the base register is to allow the data memory starting address to be moved to any location in the 4-gigabyte memory space addressed by the full 32-bit address configuration. The eight most significant address bits are supplied by the base register. After power-up or reset, the base register is loaded with >FS, where S is the board slot ID number. The memory control space used by the configuration ROM and the on-board registers is not moved by the base address register.

**NOTE:** In the BASE(07:00) code, (07:00) indicates bits 00 through 07. This method of indicating bits is followed throughout this document and is standard on the block diagram and the memory board logic diagram.

When the system is initialized on power-up, bits 03:00 of the base register are loaded with the ID(3:0)– code from the backplane, and bits 07:04 of the base register are set. This allows the system software to address 256 megabytes of memory data at address >FSXXXXXX. After all initialization is complete, the system software can address other 256-megabyte sections of memory by writing the desired address bits into the base register. This allows large memories to be addressed using all of the 32-bit address capacity of the NuBus and local bus.

The base register is loaded by writing data bits DB(07:00) at address >FSFFC008. Control space resides at >FSFFC000 through FSFFFFFF, and control register 00 resides at location >FSFFC000, regardless of the contents of the base register. The full 32-bit address configuration is shown in Figure 4-3.

**Figure 4-3** **Full 32-Bit Address Configuration**




**NOTE**

*The range of addresses for the board is >FS000000 through >FSFFFFFF immediately after power-up or board reset. With software modification the address range can be >00000000 through >FFFFFFFF.

**NuBus Structure**

**4.2.3** The NuBus is a multiplexed 32-bit bus that carries both addresses and data between the NUPI and other system devices. The bus also carries control and status information. The NUPI is allocated a 16-megabyte block in the NuBus address space. This memory block is used for all NUPI slave operations, and its location in memory is determined by the slot in which the NUPI is installed.

Figure 4-4 shows the organization of NuBus bits. These bits are designated as bits 0 through 31, where bit 0 is the least significant bit and bit 31 is the most significant bit. The NuBus can perform 32-bit word, 16-bit halfword, or 8-bit byte operations.

**Figure 4-4** **NuBus Bit Organization**

bit 31 bit 0

| 32-bit word | | | |
|---|---|---|---|
| halfword 1 | | halfword 0 | |
| byte 3 | byte 2 | byte 1 | byte 0 |

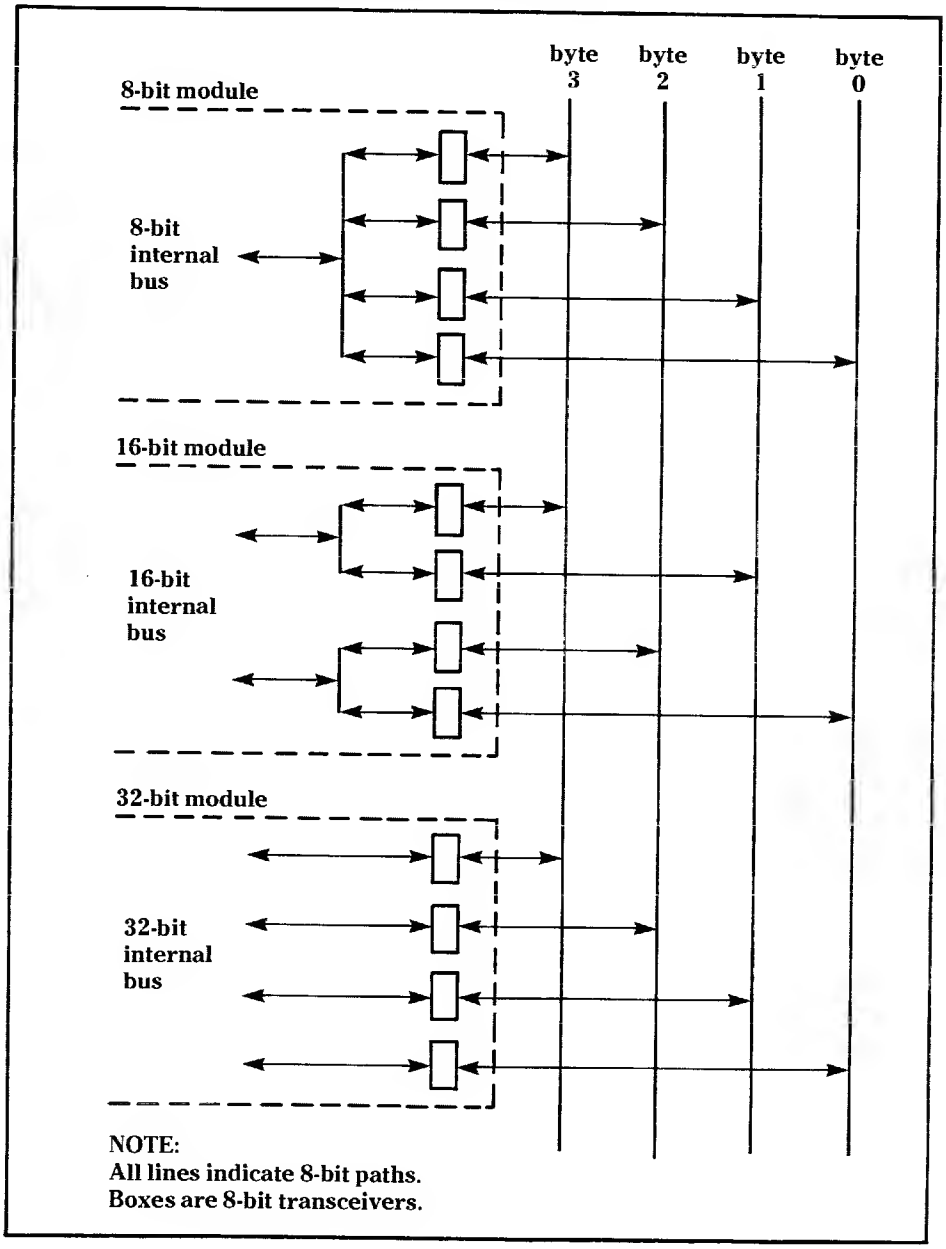
Figure 4-5 shows the data paths for 8-bit bytes, 16-bit halfwords, and 32-bit words over the NuBus. Any given data byte is always carried on the same set of NuBus lines regardless of the transfer mode. The address lines are allocated as follows:

- Byte 0 — AD0– through AD7–
- Byte 1 — AD8– through AD15–

- Byte 2 — AD16– through AD23–
- Byte 3 — AD24– through AD31–

This data path approach allows a straightforward connection of either 8-bit, 16-bit, or 32-bit devices.

**Figure 4-5** **Data Paths for Bytes, Halfwords, and Words**



### Data Transfer Mode Signals

**4.2.4** Table 4-2 defines the data transfer signals for the NuBus and the Explorer bus. Signals TM0– and TM1– are part of the NuBus. Signals LTM0– and LTM1– are part of the local bus. The address bits AD0– and AD1– are common to both the NuBus and the local bus. The actual decoding of these signals occurs at the data strobe and NuBus acquisition logic on the block diagram.

The TM0– and TM1– signals have been changed to NUTM0– and NUTM1– signals by the lock logic circuits. The data transfer mode signals are decoded to create a DS(3:0) code that is sent to the column address strobe (CAS) drivers. The CAS drivers determine which columns on the memory arrays to address for the type of read or write cycle to be used. The write and read block cycles are supported only when the memory board is a slave to the NuBus.

**Table 4-2 Data Transfer Mode Signals**

| *Signals (Low True)* | | | | |
|---|---|---|---|---|
| ***TM1– or LTM1–*** | ***TM0– or LTM0–*** | ***AD1–*** | ***AD0–*** | ***Type of Read or Write Cycle*** |
| L | L | L | L | Write byte 3 |
| L | L | L | H | Write byte 2 |
| L | L | H | L | Write byte 1 |
| L | L | H | H | Write byte 0 |
| L | H | L | L | Write halfword 1 |
| L | H | L | H | Write block* |
| L | H | H | L | Write halfword 0 |
| L | H | H | H | Write word |
| H | L | L | L | Read byte 3 |
| H | L | L | H | Read byte 2 |
| H | L | H | L | Read byte 1 |
| H | L | H | H | Read byte 0 |
| H | H | L | L | Read halfword 1 |
| H | H | L | H | Read block* |
| H | H | H | L | Read halfword 0 |
| H | H | H | H | Read word |

**NOTE:**

* The write block and read block cycles are supported only when the memory board is a slave to the NuBus.

**NuBus Status Code Signals**

**4.2.5** The NuBus status code signals are shown in Table 4-3. The TM0– and TM1– lines provide status information to the current NuBus master during the acknowledge cycle. The memory board can be a NuBus master or slave during single data-cycle transactions. Block transfers are supported only when the memory board is a NuBus slave. The status information is briefly defined in the following items:

- A *NuBus transfer complete* indicates a normal, valid completion of a bus transfer.
- An *error* can occur during a read or write operation. When an error occurs, the transaction terminates in a normal way and the NuBus master assumes the responsibility for handling the error condition.
- A *NuBus time-out* occurs when an unimplemented address location is requested. This time-out response indicates that the system-defined time-out period has transpired at the same time the NuBus was busy (between start and acknowledge cycles) and no transfer acknowledge has occurred. Bus time-out also occurs when a contender requests the NuBus and does not generate a start cycle. In this case, the NuBus time-out logic generates an idle cycle to reinitiate bus arbitration. Bus time-out responses are generated by the system time-out logic, which is not on the memory board.
- A *try again later* status code is a response from the addressed slave and indicates that it is not able to respond to a NuBus master's transfer request at the time of the request. The NuBus master will rearbitrate for the NuBus later and try again to get a response from the selected slave.

**Table 4-3** **NuBus Status Code Summary**

| *Signals* | | |
|---|---|---|
| ***TM1–*** | ***TM0–*** | ***Status Information*** |
| L | L | NuBus transfer complete |
| L | H | Error |
| H | L | NuBus time-out |
| H | H | Try again later |

### NuBus Block Sizes and Starting Addresses

**4.2.6** Table 4-4 lists the NuBus block sizes and starting addresses. Address bits ADI(5:2)– provide the code that indicates the number of words in a block and the block starting address. Address bits ADI(5:2)– are changed to a CADR(5:2)– code at the block and cycle control logic on the block diagram. The CADR(5:2)– code is sent to the refresh and multiplexer logic to become part of the complete word address. If a block transfer is not being implemented, address bits ADI(5:2)– are the same as the CADR(5:2)– code. Only 32-bit word transfers are supported in the block transfer mode.

During block transfers, the memory board is a slave to the NuBus. Block transfers are composed of a start cycle, multiple data cycles to and/or from sequential address locations, and an acknowledge cycle. For each data cycle during a block transfer, the memory board responds to the NuBus using the TM0–, TM1–, and ACK– signals. The intermediate data cycle responses are data cycles where TM0– is asserted and TM1– and ACK– are both unasserted. For intermediate responses, TM0– has the same significance and timing as the ACK– signal during nonblock transfers. The last word transfer in the acknowledge block is indicated when the ACK– signal is asserted.

**Table 4-4 NuBus Block Sizes and Starting Addresses**

| *Address Bits* | | | | *Number of* | |
|---|---|---|---|---|---|
| ***ADI05–*** | ***ADI04–*** | ***ADI03–*** | ***ADI02–*** | ***Words in Block*** | ***Block Starting Address*** |
| X | X | X | H | 2 | (AD31–AD03)000 |
| X | X | H | L | 4 | (AD31–AD04)0000 |
| X | H | L | L | 8 | (AD31–AD05)00000 |
| H | L | L | L | 16 | (AD31–AD06)000000 |

**NOTE:**

AD(31:00)– are NuBus address/data signals. After leaving the NuBus, the AD(31:00)– signals become ADI(31:00) signals on the memory board.

### Clock Generation

**4.2.7** The bus currently selected by the bus arbitration logic, either the local bus clock (BCLK–) or the NuBus clock (ICLK–) drives the board clock (BCLK) synchronously and in phase. Figure 4-6 and Figure 4-7 illustrate board clock regeneration from the local bus clock and the NuBus clock, respectively.

Figure 4-6 **Board Clock Regeneration From the Local Bus Clock**

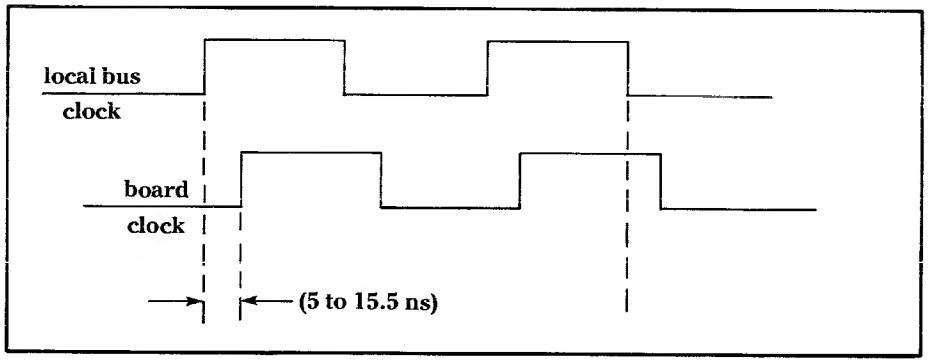
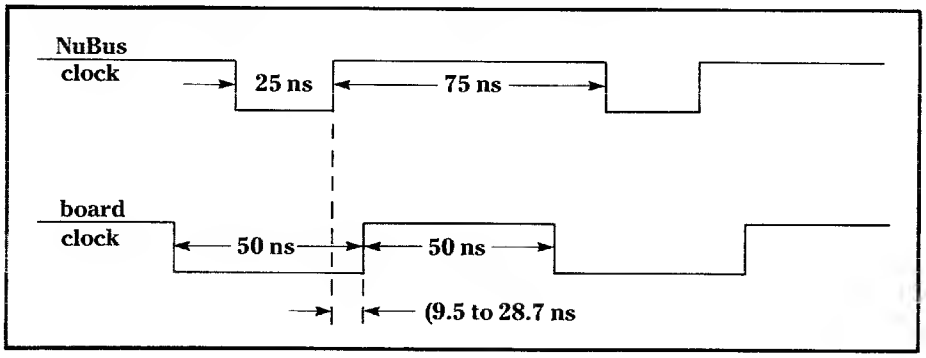



Figure 4-7 **Board Clock Regeneration From the NuBus Clock**




**Bus Arbitration Logic**

**4.2.8** The bus arbitration selects between NuBus and local bus cycles. The arbitration gives each bus an equal share of the memory board resources. When both buses want access to the board, the one currently selected gets access. After completion of the current cycle, the competing bus is selected.

After initial power-up or a board reset, the NuBus clock is selected to drive the board clock.

**NuBus Arbitration**

**4.2.9** When a NuBus arbitration contest occurs, each board that is a contender drives the ARB(3:0)– lines with its unique ID code. If a board detects a higher ID code than its own on the ARB(3:0)– lines, it removes its ARB(3:0)– code. Then, only the board with the highest ID code is master of the bus.

Arbitration contests last two clock periods by definition. On the second sample edge of the clock after a contest starts, all contenders test their internal GRANT signal. The highest priority contender has its GRANT signal asserted. The winning contender acquires control of the bus and asserts the START– signal on the next assertion edge of the clock. If the bus is in use, the new winner asserts the START– signal on the assertion edge immediately after the next sample edge where the current transaction's ACK– signal is asserted. The new winner continues to assert its ID code on the ARB(3:0)– lines during the start cycle of its first transaction to facilitate bus lock detection and bus diagnostics.

**Control Logic**

**4.2.10** There are four control logic blocks on the memory board block diagram. These are the buffer control logic, refresh control logic, block cycle control logic, and register control logic.

The buffer control logic arbitrates between NuBus to memory requests, local bus to memory requests, local bus to NuBus requests, and refresh cycles. The arbitration between the local bus and the NuBus gives each bus an equal share of the memory board resource. If both buses try to access the memory board at the same time, the bus currently pointed to by the buffer control logic gets access. After completion of the current cycle, the direction changes to allow access from the other bus.

The refresh and control logic, block cycle and control logic, and the register control logic functions are reasonably self-evident from the block diagram.

**Local Bus and NuBus Locking**

**4.2.11** Bus locking is a mechanism used to develop continuous read/write cycles on either the local bus or the NuBus. The local bus is locked to the memory board when the LOCK– signal is asserted. The NuBus is locked when a master continues to request and contend for the bus. Because the master in this case has the highest ID code of the boards present, it wins arbitration contests that follow.

**Jumper Options**

**4.2.12** The memory board is designed for either 64K-bit or 256K-bit memory chips. Table 4-5 shows the jumper organization for 64K-bit and 256K-bit memory chips. The table also shows the jumper for a memory board fully equipped or half equipped with memory chips. These jumpers are hard-wired, zero-ohm resistors installed at the factory. The information given here is to help understand how the memory board can use either size memory chip. For physical locations of the jumpers, check the memory board assembly drawing.

Jumpers AD049 and AK069 are electrically part of the data strobe and NuBus acquisition logic. All the other jumpers are electrically part of the row address strobe (RAS) driver jumper options logic. The word *in* indicates the jumper is installed. The word *out* indicates the jumper is not installed.

**Table 4-5 Jumpers for 64K-Bit and 256K-Bit chips**

| *Memory Size* | *Jumpers* | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ***AD049*** | ***AK069*** | ***CC074*** | ***CC079*** | ***AK064*** | ***CC064*** | ***BG064*** | ***BG044*** |
| 64K-bit chips | | In | In | In | Out | Out | Out | Out |
| 256K-bit chips | | Out | Out | Out | In | In | In | In |
| Full board | In | | | | | | | |
| Half board | Out | | | | | | | |

**NOTE:**

Jumpers are factory-installed, hard-wired, zero-ohm resistors.

To locate the jumpers, physically, look at the board oriented as shown in Figure 1-1. AD, AK, and so on are row designations at the sides of the board. Vertical column positions are numbered horizontally at the top of the board: 03, 07, 11, and so on.

**Memory Address Development**

**4.2.13** Table 4-6 and Table 4-7 show the address development of memory boards equipped with 64K-bit and 256K-bit memory chips. A memory board with 64K-bit memory chips has a capacity of 2 megabytes, requiring address bits ADI(20:00). A memory board with 256K-bit memory chips has a capacity of 8 megabytes, requiring address bits ADI(22:00). The actual addressing is broken into three code groups:

- Row address strobe (RAS) code bits
- Multiplexed address (MUX) code bits
- Column address strobe (CAS) code bits

These three code groups function together in the RAS/MUX/CAS sequence to address the memory chips. The RAS/MUX/CAS sequence is explained in the following paragraph.

For a memory board with 64K-bit memory chips, address bits ADI(20:18) generate the ASEL(2:0) code that enables the RAS drivers. Address bits ADI(17:06) and bits CADR(05:02) are multiplexed to generate address driver code ADDR(07:00), which drives each of the eight address buffers. The output of the eight address buffers provides the address for eight memory chips and one parity memory chip on one physical row of memory. Address bits ADI(01:00) and transfer mode bits TM(1:0) generate a DS(3:0) code that enables the CAS drivers. Data strobe code DS(3:0) determines if bytes, halfwords, or 32-bit words are addressed.

The RAS/MUX/CAS sequence for a memory board with 256K-bit memory chips is similar to the sequence for 64K-bit memory chips, except for additional address bits as indicated in Table 4-7.

**Table 4-6** **Address Development for 64K-Bit Memory Chips**

| *Item* | *RAS Code* | *Multiplexed Address* | *CAS Code* |
|---|---|---|---|
| Address Bits | ADI(20:18) | ADI(17:06) CADR(05:02) | ADI(01:00) TM(1:0) |
| Output Signal | ASEL(2:0) | ADDR(07:00) | DS(3:0) |

**Table 4-7** **Address Development for 256K-Bit Memory Chips**

| *Item* | *RAS Code* | *Multiplexed Address* | *CAS Code* |
|---|---|---|---|
| Address Bits | ADI(22:20) | ADI(19:06) CADR(05:02) | ADI(01:00) TM(1:0) |
| Output Signal | ASEL(2:0) | ADDR(08:00) | DS(3:0) |

**Memory Refresh Cycle**

**4.2.14** The refresh logic uses an on-board frequency counter that counts down the NuBus 10-megahertz clock to generate refresh requests. The refresh timer signals the control logic that a refresh cycle is required. Normally a refresh cycle starts immediately if the memory board is idle, or after the current memory cycle if the board is busy.

Refresh cycles compete with the central processing unit (CPU) cycles for memory board control. If both the CPU and refresh cycles seek access to

the memory board at the same time, the CPU cycle gets preference. Extended cycles, similar to those caused by processor single stepping, can cause the memory cycle to terminate abnormally due to the necessity of refreshing the memory chips each 15 microseconds. Memory data integrity is maintained but data read back to the processor on the local bus is not guaranteed.

**Signal Definitions**

**4.2.15** Local bus and NuBus signal definitions are given in Table 4-8 and Table 4-9. Other signals mentioned in this manual are listed in Table 4-10.

**Table 4-8 Local Bus Signal Definitions**

| *Signal Signature* | *Definition* |
|---|---|
| BCLK–* | This is a 50 percent duty cycle signal generated by the processor board on the local bus and used on the memory board to generate other clock signals. It has a frequency range of 7 to 10 megahertz. If the BCLK– clock signal is absent, each memory board regenerates a 50 percent duty cycle BCLK– clock signal from the NuBus clock. The NuBus clock signal is used for all the memory board circuits related to the NuBus, while the BCLK– signal is the clock for memory board circuits not related to the NuBus. |
| MEMREQ– | This is a memory request signal that is generated by the processor on the local bus. When MEMREQ– is asserted, it tells the memory board that a memory cycle is required. MEMREQ– is checked on the rising edge of the local bus clock. |
| MEMACK– | This is a memory acknowledge signal that is generated by the memory board. Data and error information will be valid on the local bus at the next falling edge of the local bus clock after one of the following two events, whichever occurs last:<br>■ When 284 nanoseconds have passed after the falling edge of the local bus clock that signified the start of the access.<br>■ When MEMACK– becomes true. |
| BERR– | This is a bus error signal that is generated by the memory board. When BERR– is asserted, it tells the processor on the local bus that an error was detected during the data transfer. |
| LAD(31:00)– | These are the 32-bit address signals that are generated by the processor on the local bus. |

**Table 4-8** **Local Bus Signal Definitions (Continued)**

| *Signal Signature* | *Definition* |
|---|---|
| DAT(31:00)– | These are the 32-bit data signals that can originate at the processor on the local bus, the memory board, or any master on the NuBus. |
| LTM0;–, LTM1– | These are local bus transfer mode signals that are generated by the processor on the local bus. LTM0– and LTM1– are used with LAD00– and LAD01– to form a 4-bit binary code that tells the memory board the type of transfer mode the processor is requesting. The transfer modes available on the local bus are read and write words, halfwords, or bytes. |
| BS0– | This is a local bus select signal generated by the processor on the local bus. BS0– is a line from the processor that is wired to only one of the memory boards. When BS0– is asserted, it tells the specific memory board to which it is wired that it is the NuBus master. |
| DECODE– | This is a decode signal that is generated by the memory board. Any memory board currently accessed by the local bus asserts the DECODE– signal. The NuBus master on the memory board uses this signal to determine if any memory board is being accessed by the local bus. If the local bus is not accessing any memory board, the NuBus master on the memory board can access the NuBus and complete a memory cycle. |
| FAST– | Although this signal exists on the backplane, it is not used by the memory board. |
| LOCK– | This signal is generated by the processor on the local bus. When LOCK– is asserted, it tells the memory board that is being accessed that the on-board memory is locked and cannot be accessed by the NuBus. When locked, the memory board stays locked until LOCK– is negated or unasserted. |
| BIGRD– | This signal is used for factory test. When BIGRD– is asserted, the read and write parity bits, PEEKIN(3:0) and PEEKOUT(3:0), are available at connector P3. |

**NOTE:**

* The local bus clock (BCLK–) is a different signal from the board clock (BCLK or its inversion BCLK–).

**Table 4-9 NuBus Signal Definitions**

| *Signal Signature* | *Definition* |
|---|---|
| ICLK– | This clock signal originates at the system interface board. Its main function is to synchronize bus arbitration and data transfers between system modules on the NuBus. This clock signal has a nominal frequency of 10 megahertz with a duty cycle of 75 percent. In general, signals are changed at the rising edge of ICLK– and tested at the falling edge. |
| START– | This is a transfer start signal that is asserted for only one clock period by the memory board bus master at the beginning of a data transfer operation. START– indicates to the slaves on the NuBus that the address/data signals are carrying a valid address. |
| ACK– | This is a transfer acknowledge signal that is asserted for one clock period by the memory board when it is a slave to the NuBus. ACK– indicates completion of a data transfer operation to the memory board from the NuBus. |
| TM0–, TM1– | These are transfer mode signals that are asserted by the memory board bus master during start cycles to indicate the type of data transfer being started. These signals are also asserted during acknowledge cycles by the memory board when it is a slave to the NuBus to indicate the type of acknowledgment sent to modules on the NuBus. |
| RESET– | The reset signal has two functions determined by its interval. When RESET– is asserted for a single clock period, it causes the interface (bus reset) of all boards on the NuBus to be initialized. When RESET– is asserted for more than one clock period, it returns all boards on the NuBus to their initial power-up state (system reset). |
| AD0–, AD1– | These are address signals that carry address information during the start cycle of a byte transfer. AD0– and AD1– carry control information during the start cycle of a word or halfword transfer. |
| AD(31:00)– | These are NuBus address/data signals that are multiplexed to carry a 32-bit address at the start of a cycle and a 32-bit data code during the remainder of the cycle. Signals AD(31:00)– can be generated by any master on the NuBus, including the memory board. |
| RQST– | This is a bus request signal that is asserted by any master on the NuBus that wants control of the bus. The memory board can assert this signal when it is a NuBus master. |

**Table 4-9** **NuBus Signal Definitions (Continued)**

| *Signal Signature* | *Definition* |
|---|---|
| ARB(3:0)– | These signals form an arbitration binary code that is used by the distributed arbitration logic to determine which component is the NuBus master. Signal code ARB(3:0)– can be generated by any master on the NuBus, including the memory board. |
| ID(3:0)– | These signals form a slot identification binary code that is used to identify each board by slot location. The ID(3:0)– lines are wired to ground on the backplane to provide a separate code for each of slots 0 through 15. |
| SP–, SPV– | These are system parity signals that are used by some modules on the NuBus. They are not used by the memory board. |

**Table 4-10** **Additional Explorer Signal Definitions**

| *Signal Signature* | *Definition* |
|---|---|
| ADI(00:31) | This is an internal, on-board, 32-bit address bus on which NuBus or local bus addresses are latched. |
| ADDR(00:08) | This is a 9-bit address bus output from the RAS/CAS address multiplexers for input to the memory chip. |
| ASEL(00:02) | The address select bus is made up of three address lines which are generated from the AD1 bus according to the type of memory chips used on the board. These address lines are decoded to generate the appropriate memory row enable signals. |
| CADR(02:05) | The CAS address bus is a 4-bit bus output from the block control programmable array logic (PAL). They are the least significant bits input to the address multiplexers. |
| CAS | This is the acronym for column address strobe. The RAS/MUX/CAS sequence is explained in the paragraph entitled Memory Address Development. |
| DAT (00:31) | This is a 32-bit data bus that is a part of the local bus. |
| DS (00:03) | The data strobe bus is a 4-bit bus output from the byte strobe decode PAL. These bits enable the CAS drivers. |

**Table 4-10** **Additional Explorer Signal Definitions (Continued)**

| *Signal Signature* | *Definition* |
|---|---|
| GRANT | This signal indicates that the memory board has gained control of the NuBus after the arbitration period. |
| MUX | This is the abbreviation for multiplexer. The RAS/MUX/CAS sequence is explained in the paragraph entitled Memory Address Development. |
| NUTM0– and NUTM1– | NUTM0– and NUTM1– are on-board signals that are generated by latching the NuBus TM0– and TM1– signals when a NuBus cycle is started. |
| PEEKIN (00:03) | PEEKIN signals from connector P3 are used to input parity data from an external source for diagnostic testing. |
| PEEKOUT (00:03) | PEEKOUT signals from connector P3 are used to read internal parity during diagnostic testing. |
| RAS | This is an acronym for row address strobe. The RAS/MUX/CAS sequence is explained in the paragraph entitled Memory Address Development. |

**Connector Pin/ Signal Assignments**

**4.2.16** Connector pin numbers and signal assignments are related to the numbers in parentheses, from (01) through (96), that are stamped on the memory board. The physical appearance of connector pin rows C, B, and A (from left to right when viewed from the back of the system enclosure) is the same as on the backplane. Table 4-11 through Table 4-13 show the relation between the connector pin numbers, signals, memory board numbers, and the directional response of the signals.

## Table 4-11 NuBus Connector P1 Signals

| *Connector Pin No.* | *Row C* | *Signal* | *I/O* | *Row B* | *Signal* | *I/O* | *Row A* | *Signal* | *I/O* |
|---|---|---|---|---|---|---|---|---|---|
| 1 | (65) | RESET– | O | (33) | –12 | I | (01) | –12 | I |
| 2 | (66) | GND | | (34) | GND | | (02) | GND | |
| 3 | (67) | +5V | I | (35) | GND | | (03) | SPV– | |
| 4 | (68) | +5V | I | (36) | +5V | I | (04) | SP– | |
| 5 | (69) | TM0– | I/O | (37) | +5V | I | (05) | TM1– | I/O |
| 6 | (70) | AD0– | I/O | (38) | +5V | I | (06) | AD1– | I/O |
| 7 | (71) | AD2– | I/O | (39) | +5V | I | (07) | AD3– | I/O |
| 8 | (72) | AD4– | I/O | (40) | –5V | I | (08) | AD5– | I/O |
| 9 | (73) | AD6– | I/O | (41) | –5V | I | (09) | AD7– | I/O |
| 10 | (74) | AD8– | I/O | (42) | –5V | I | (10) | AD9– | I/O |
| 11 | (75) | AD10– | I/O | (43) | –5V | I | (11) | AD11– | I/O |
| 12 | (76) | AD12– | I/O | (44) | GND | | (12) | AD13– | I/O |
| 13 | (77) | AD14– | I/O | (45) | GND | | (13) | AD15– | I/O |
| 14 | (78) | AD16– | I/O | (46) | GND | | (14) | AD17– | I/O |
| 15 | (79) | AD18– | I/O | (47) | GND | | (15) | AD19– | I/O |
| 16 | (80) | AD20– | I/O | (48) | GND | | (16) | AD21– | I/O |
| 17 | (81) | AD22– | I/O | (49) | GND | | (17) | AD23– | I/O |
| 18 | (82) | AD24– | I/O | (50) | GND | | (18) | AD25– | I/O |
| 19 | (83) | AD26– | I/O | (51) | GND | | (19) | AD27– | I/O |
| 20 | (84) | AD28– | I/O | (52) | GND | | (20) | AD29– | I/O |
| 21 | (85) | AD30– | I/O | (53) | GND | | (21) | AD31– | I/O |
| 22 | (86) | GND | | (54) | GND | | (22) | GND | |
| 23 | (87) | GND | | (55) | GND | | (23) | GND | |
| 24 | (88) | ARB0– | O | (56) | –5V | I | (24) | ARB1– | O |
| 25 | (89) | ARB2– | O | (57) | –5V | I | (25) | ARB3– | O |
| 26 | (90) | ID0– | I | (58) | –5V | I | (26) | ID1– | I |
| 27 | (91) | ID2– | I | (59) | –5V | I | (27) | ID3– | I |
| 28 | (92) | START– | I/O | (60) | +5V | I | (28) | ACK– | I/O |
| 29 | (93) | +5V | I | (61) | +5V | I | (29) | +5V | I |
| 30 | (94) | +5V | I | (62) | GND | | (30) | RQST– | I/O |
| 31 | (95) | GND | | (63) | GND | | (31) | GND | |
| 32 | (96) | ICLK– | I | (64) | +12V | I | (32) | +12V | I |

**NOTE:**

Signals SPV– and SP– are not connected on the memory board.

## Table 4-12 Local Bus Connector P2 Signals

| *Connector Pin No.* | *Row C* | *Signal* | *I/O* | *Row B* | *Signal* | *I/O* | *Row A* | *Signal* | *I/O* |
|---|---|---|---|---|---|---|---|---|---|
| 1 | (65) | DAT00– | I/O | (33) | LTM0– | I | (01) | AD00– | I |
| 2 | (66) | DAT01– | I/O | (34) | GND | | (02) | AD01– | I |
| 3 | (67) | DAT02– | I/O | (35) | GND | | (03) | AD02– | I |
| 4 | (68) | DAT03– | I/O | (36) | | | (04) | AD03– | I |
| 5 | (69) | DAT04– | I/O | (37) | +5V | I | (05) | AD04– | I |
| 6 | (70) | DAT05– | I/O | (38) | +5V | I | (06) | AD05– | I |
| 7 | (71) | DAT06– | I/O | (39) | BIGRD– | I | (07) | AD06– | I |
| 8 | (72) | DAT07– | I/O | (40) | SPARE 1 | | (08) | AD07– | I |
| 9 | (73) | DAT08– | I/O | (41) | MEMREQ– | I | (09) | AD08– | I |
| 10 | (74) | DAT09– | I/O | (42) | LOCK– | I | (10) | ADD9– | I |
| 11 | (75) | DAT10– | I/O | (43) | | | (11) | AD10– | I |
| 12 | (76) | DAT11– | I/O | (44) | GND | | (12) | AD11– | I |
| 13 | (77) | DAT12– | I/O | (45) | BS0– | I | (13) | AD12– | I |
| 14 | (78) | DAT13– | I/O | (46) | MEMACK– | O | (14) | AD13– | I |
| 15 | (79) | DAT14– | I/O | (47) | | | (15) | AD14– | I |
| 16 | (80) | DAT15– | I/O | (48) | GND | | (16) | AD15– | I |
| 17 | (81) | DAT16– | I/O | (49) | | | (17) | AD16– | I |
| 18 | (82) | DAT17– | I/O | (50) | | | (18) | AD17– | I |
| 19 | (83) | DAT18– | I/O | (51) | GND | | (19) | AD18– | I |
| 20 | (84) | DAT19– | I/O | (52) | | | (20) | AD19– | I |
| 21 | (85) | DAT20– | I/O | (53) | BCLK– | | (21) | AD20– | I |
| 22 | (86) | DAT21– | I/O | (54) | BERR– | O | (22) | AD21– | I |
| 23 | (87) | DAT22– | I/O | (55) | GND | | (23) | AD22– | I |
| 24 | (88) | DAT23– | I/O | (56) | FAST– | I | (24) | AD23– | I |
| 25 | (89) | DAT24– | I/O | (57) | | | (25) | AD24– | I |
| 26 | (90) | DAT25– | I/O | (58) | DECODE– | O | (26) | AD25– | I |
| 27 | (91) | DAT26– | I/O | (59) | SPARE 2 | | (27) | AD26– | I |
| 28 | (92) | DAT27– | I/O | (60) | +5V | I | (28) | AD27– | I |
| 29 | (93) | DAT28– | I/O | (61) | | | (29) | AD28– | I |
| 30 | (94) | DAT29– | I/O | (62) | GND | | (30) | AD29– | I |
| 31 | (95) | DAT30– | I/O | (63) | GND | | (31) | AD30– | I |
| 32 | (96) | DAT31– | I/O | (64) | LTM1– | I | (32) | AD31– | I |

**NOTES:**

The BIGRD– signal is +5 V on the local bus.

BCLK– is the local bus clock signal.

## Table 4-13 I/O Connector P3 Signals

| *Connector Pin No.* | *Row C* | *Signal* | *I/O* | *Row B* | *Signal* | *I/O* | *Row A* | *Signal** | *I/O** |
|---|---|---|---|---|---|---|---|---|---|
| 1 | (65) | PEEKIN3 | I | (33) | | | (01) | | |
| 2 | (66) | PEEKIN2 | I | (34) | GND | | (02) | | |
| 3 | (67) | PEEKIN1 | I | (35) | GND | | (03) | | |
| 4 | (68) | PEEKIN0 | I | (36) | | | (04) | | |
| 5 | (69) | PEEKOUT3 | O | (37) | +5V | I | (05) | | |
| 6 | (70) | PEEKOUT2 | O | (38) | +5V | I | (06) | | |
| 7 | (71) | PEEKOUT1 | O | (39) | +5V | I | (07) | | |
| 8 | (72) | PEEKOUT0 | O | (40) | | | (08) | | |
| 9 | (73) | | | (41) | | | (09) | | |
| 10 | (74) | | | (42) | | | (10) | | |
| 11 | (75) | | | (43) | | | (11) | | |
| 12 | (76) | | | (44) | GND | | (12) | | |
| 13 | (77) | | | (45) | | | (13) | | |
| 14 | (78) | | | (46) | | | (14) | | |
| 15 | (79) | | | (47) | | | (15) | | |
| 16 | (80) | | | (48) | GND | | (16) | | |
| 17 | (81) | | | (49) | | | (17) | | |
| 18 | (82) | | | (50) | | | (18) | | |
| 19 | (83) | | | (51) | GND | | (19) | | |
| 20 | (84) | | | (52) | | | (20) | | |
| 21 | (85) | | | (53) | | | (21) | | |
| 22 | (86) | | | (54) | | | (22) | | |
| 23 | (87) | | | (55) | GND | | (23) | | |
| 24 | (88) | | | (56) | | | (24) | | |
| 25 | (89) | | | (57) | | | (25) | | |
| 26 | (90) | | | (58) | | | (26) | | |
| 27 | (91) | | | (59) | | | (27) | | |
| 28 | (92) | | | (60) | +5V | | (28) | | |
| 29 | (93) | | | (61) | | | (29) | | |
| 30 | (94) | | | (62) | GND | | (30) | | |
| 31 | (95) | | | (63) | GND | | (31) | | |
| 32 | (96) | | | (64) | | | (32) | | |

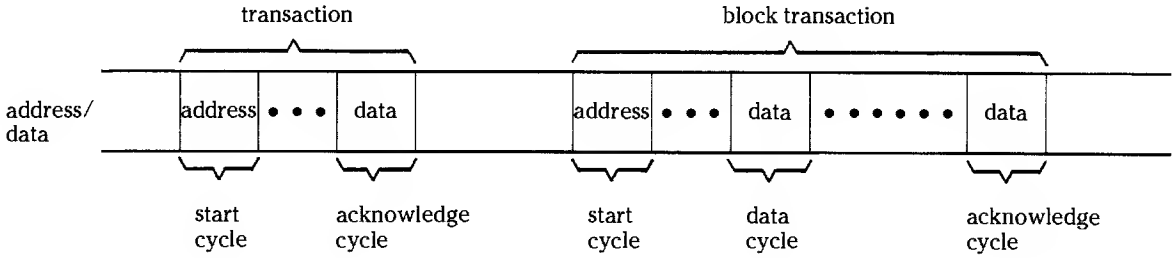
**NOTE:**

* Row A has no signals or I/O information on P3.

## Data Transfer Operation

4.3 The memory board supports single data cycle transactions on both the NuBus and the local bus. Block data transfers are supported by the memory board when it is a slave to the NuBus. Figure 4-8 shows the relationship between single cycle transactions and block transactions. A brief analysis of single data cycle transactions, block data transfers, and interrupt operations is given in the following paragraphs.

**Figure 4-8 Single Cycle and Block Transaction Comparisons**




### Single Data Cycle Transaction

**4.3.1** Single data cycle transactions are made on both the NuBus and the local bus. These transactions move one data item at a time on the NuBus or local bus and are synchronized to the applicable NuBus or local bus clock. All cycle transactions on the NuBus are originated by the START– signal and ended by the ACK– signal. On the local bus, all cycle transactions are started by the MEMREQ– signal and ended on the next falling edge of the local bus clock after the specified access time. Local bus timing requirements are illustrated in Figure 4-9.

#### *Read/Write Transaction*

**4.3.1.1** A read transaction is distinguished from a write transaction by the code developed by the TM0–, TM1–, AD00–, and AD01– signals on the NuBus and the LTM0–, LTM1–, LAD00–, and LAD01– signals on the local bus. These signals also determine if a byte, halfword, or 32-bit word is to be the read or write transaction.

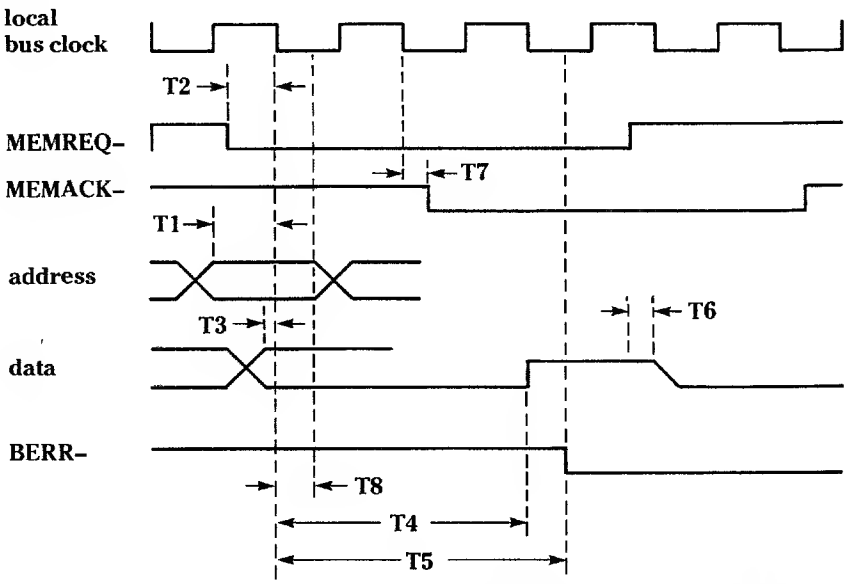
#### *Idle Cycle Operations*

**4.3.1.2** Idle cycle operations are defined as the assertion of the START– and ACK– signals in the same clock period. Idle cycles are used solely on the NuBus to reinitiate bus arbitration.

#### *Acknowledgment*

**4.3.1.3** Acknowledgment on the NuBus indicates the end of a transaction and also allows some types of status information to be passed over the TM0– and TM1– lines from the slave to the master. Acknowledgment on the Explorer bus simply means the end of a transaction.

**Figure 4-9** **Local Bus Timing Requirements**




| | Minimum | Typical | Maximum |
|---|---|---|---|
| T1 Local bus address setup | 50 | — | — |
| T2 MEMREQ– setup | 50 | — | — |
| T3 Data setup (write) | 00 | — | — |
| T4 Data access (read) | 179 | 220 | 250 |
| T5 Bus error valid | 187 | — | 284 |
| T6 Data hold after MEMREQ– | 00 | — | 35 |
| T7 MEMACK– from clock | 14 | — | 44 |
| T8 Local bus address hold | 50 | — | — |

**Block Data Transfers**

**4.3.2** Block data transfers are possible only when the memory board is a slave to the NuBus. Block transfers are composed of a start cycle, multiple data cycles to and from sequential address locations, and an acknowledge cycle. The number of data cycles is controlled by the master and communicated during the start cycle. The allowed lengths of block transfers are 2, 4, 8, and 16 words. Only 32-bit words are supported in the block transfer mode.

***Block Read/Write Transactions***

**4.3.2.1** Block read transactions are distinguished from block write transactions by the code developed by the TM0–, TM1–, AD00–, and AD01– signals on the NuBus. Because only word transactions are supported in the block mode, there is only one block read code and only one block write code. During a block read, the NuBus master starts the transfer. The responding slave, in this case the memory board, drives the data onto the bus and the NuBus master accepts the data on each intermediate or last ACK– signal. Block writes are similar to block reads except the bus master drives the data bus as the slave accepts data.

***Block Transfer Errors***

**4.3.2.2** Block transfers can be cut short by an error acknowledgment from the NuBus slave at any time. These errors are indicated by the standard NuBus status code summary shown in Table 4-3.

**Interrupt Operations**

**4.3.3** Interrupts on the NuBus and local bus are implemented as write transactions. Interrupt operations require no unique signals or protocols and are software specified by memory mapping the priority levels.

## Programming Information

**4.4** This section briefly describes the following configuration and register data:

- Control address space
- Configuration ROM
- Configuration register
- Base register
- Test register
- Failure location latch word
- NuBus termination status and error latch register

**Control Address Space**

**4.4.1** Control address space resides at >FSFFCXXX, regardless of the contents of the base register. Reading unimplemented areas of the control space can result in a board error or time-out cycle termination.

## Configuration ROM Description

**4.5** The configuration read-only memory (ROM) data resides in the top part of the on-board control address space. The configuration ROM data is accessed by addresses >FSFFFFFF to >FSFFE000. The S designation in the addresses is a four bit code that identifies the slot that a circuit board occupies. Table 4-14 shows the contents of the configuration ROM.

All data from the configuration ROM is received in byte 0 of data lines DB(07:00). Since the ROM data is one byte wide, every fourth NuBus address will be a valid ROM address, for example, FS000000, FS000004, FS000008, and so on.

**Table 4-14** **Configuration ROM Contents**

| *Item* | *Address* | *Contents Description* |
|---|---|---|
| Serial number (corresponds to the bar code markings) | FSFFFFFC | The week of manufacture expressed by 2 binary coded decimal characters from 01 through 52. |
| | FSFFFFF8 | The year of manufacture expressed by 1 ASCII character from 3 (for 1983) through 9 (for 1989) and A (for 1990) through Z (for 2010), omitting letters G, I, J, O, and Q. |
| | FSFFFFF4<br>FSFFFFF0<br>FSFFFFEC | A site identification code expressed by 3 ASCII characters, omitting letters G, I, J, O, and Q. The most significant character (as A in AUS) is located in the higher address. |
| | FSFFFFE8<br>FSFFFFE4<br>FSFFFFE0 | A sequence number that identifies all assemblies built in one week expressed as a right justified 5-character hexadecimal code. |
| | FSFFFFDC | A weighted check sum of the 11 numeric digits in the above 8 bytes expressed by 2 hexadecimal characters derived as follows:<br>Convert each digit to a weighted number (Y1 through Y11) and then calculate the check sum using the formula:<br>$(1 \times Y1)+(2 \times Y2)...(11 \times Y11)/31$<br>to get a remainder from 0 to 30. The character with that weighted number is the check character: |

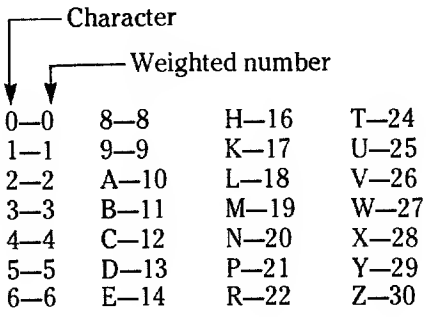


**Table 4-14 Configuration ROM Contents (Continued)**

| *Item* | *Address* | *Contents Description* |
|---|---|---|
| Configuration ROM revision levels | FSFFFFD8<br>FSFFFFD4<br>FSFFFFD0<br>FSFFFFCC<br>FSFFFFC8<br>FSFFFFC4<br>FSFFFFC0 | Sixth revision (1 ASCII character)<br>Fifth revision (1 ASCII character)<br>Fourth revision (1 ASCII character)<br>Third revision (1 ASCII character)<br>Second revision (1 ASCII character)<br>First revision (1 ASCII character)<br>Original release (1 ASCII character)<br>An asterisk (*) indicates no revision letter. The revision level is always treated as a two-character field:<br>■ Original release is at C0 and C0<br>■ First revision is at C0 and C4<br>■ Sixth revision is at C0 and C8 |
| CRC signature | FSFFFFBC<br>FSFFFFB8 | Contains 4 hexadecimal characters. The most significant character (MSC) is at FSFFFFBC. |
| Configuration ROM size | FSFFFFB4 | This field contains >0B, where B is the exponent of a base 2 logarithm, resulting in 2 raised to the 10th power, which is 2048 bytes, or 2K bytes. |
| Vendor ID | FSFFFFB0<br>FSFFFFAC<br>FSFFFFA8<br>FSFFFFA4 | Contains 4 ASCII characters.<br>TIAU indicates TI Austin.<br>LMI indicates Lisp Machine Inc.<br>Leftmost character is at FSFFFFA4. |
| Board type (binary) | FSFFFFA0 (byte 7) | Starting address: 00000000 |
| | FSFFFF9C (byte 6) | Always zero: 00000000 |
| | FSFFFF98 (byte 5) | Block size* |
| | FSFFFF94 (byte 4) | This byte contains a weighted log2 value of the gap between noncontiguous blocks of memory. This address contains 00000000. |
| | FSFFFF90 (byte 3) | Memory size* |
| | FSFFFF8C (byte 2) | Rightmost character: M (ASCII) |
| | FSFFFF88 (byte 1) | Rightmost character: E (ASCII) |
| | FSFFFF84 (byte 0) | Leftmost character: M (ASCII) |

**Table 4-14 Configuration ROM Contents (Continued)**

| *Item* | *Address* | *Contents Description* | |
|---|---|---|---|
| Part number of memory board | FSFFFF80 | Dash number — LSC | as required |
| | FSFFFF7C | ↕ | 0 |
| | FSFFFF78 | ↕ | 0 |
| | FSFFFF74 | Dash number — MSC | 0 |
| | FSFFFF70 | Hyphen | — |
| | FSFFFF6C | Part number — LSC | 0 |
| | FSFFFF68 | ↕ | 1 |
| | FSFFFF64 | ↕ | 9 |
| | FSFFFF60 | ↕ | 3 |
| | FSFFFF5C | ↕ | 4 |
| | FSFFFF58 | ↕ | 2 |
| | FSFFFF54 | ↕ | 2 |
| | FSFFFF50 | ↕ | 0 |
| | FSFFFF4C | ↕ | 0 |
| | FSFFFF48 | ↕ | 0 |
| | FSFFFF44 | Part number — MSC | 0 |
| Configuration register offset | FSFFFF40<br>FSFFFF3C<br>FSFFFF38 | Contains 6 hexadecimal characters. The address is FS000000 plus offset. The MSC is at FSFFFF40. The offset for the memory board is >FFC000. | |
| Device driver offset | FSFFFF34<br>FSFFFF30<br>FSFFFF2C | This field does not apply to the Explorer memory board. | |
| Diagnostic offset | FSFFFF28<br>FSFFFF24<br>FSFFFF20 | Contains 6 hexadecimal characters. The address is FS000000 plus offset. The MSC is at FSFFFF28. The offset for the memory board is >FFE000. | |
| Flag register offset | FSFFFF1C<br>FSFFFF18<br>FSFFFF14 | This field does not apply to the Explorer memory board. | |
| ROM flag | FSFFFF10 | Contains 8 binary bits that identify system information as follows:<br>Bit 0 — 1 = Executes self-test<br>0 = Does not execute self-test<br>Bit 1 — 1 = Executes NuBus tests<br>0 = Does not execute NuBus tests<br>Bit 2 — 1 = Can be boot master<br>0 = Cannot be boot master<br>Bit 3 — 1 = Has block support<br>0 = Has no block support | |

**Table 4-14 Configuration ROM Contents (Continued)**

| *Item* | *Address* | *Contents Description* |
|---|---|---|
| | | Bit 4 — 1 = Has system memory<br>0 = Has no system memory<br>Bit 5 — 1 = Requires a power failure warning event<br>0 = Requires warning event<br>Bit 6 — Reserved<br>Bit 7 — Reserved<br>This location on the memory board contains > 18 |
| Layout byte | FSFFFF0C | Contains 2 hexadecimal characters that indicate the ROM revision level. Updated each time the configuration ROM format changes. |
| Test time | FSFFFF08 | Contains 2 hexadecimal characters that represent a weighted log2 value. This is the extended self-test time limit in seconds. The standard self-test executes in 20 seconds or less. The standard self-test for STBM executes in 10 seconds or less. |
| ID byte | FSFFFF04 | Contains hexadecimal characters C3, which is a known value for the configuration ROM verification. |
| Resource type | FSFFFF00 | Contains 8 binary bits that identify resources by the bits that are set to 1 as follows:<br>Bit 0 — Memory<br>Bit 1 — Boot source<br>Bit 2 — LAN<br>Bit 3 — Monitor<br>Bit 4 — Bootable processor<br>Bit 5 — Keyboard<br>Bit 6 — NVRAM<br>Bit 7 — Sub-boards<br>This address on the memory board contains 00000001. |
| | FSFFFEFC through FSFFFEAC | FSFFFEFC through FSFFFEAC do not apply to the memory board. |

**Table 4-14 Configuration ROM Contents (Continued)**

| *Item* | *Address* | *Contents Description* |
|---|---|---|
| Diagnostic code | FSFFFEA8 through FSFFE000 | Diagnostic firmware for self-tests. |

**NOTES:**

A weighted log2 value is a coding scheme in which the most significant digit is a hexadecimal mantissa (M) and the least significant digit is an exponent (E) to the base 2. The value is M*(2**E).

The ROM is accessed by byte only, not words. Valid header addresses are >FSFFFF00, −04, −08,....., −F4, F8, FC. ROM data is valid only on the least significant byte of NuBus.

The binary fields are stored such that the logically highest NuBus address of each field contains the most significant byte, while the lowest address contains the least significant byte.

ASCII fields are stored as strings, with the first (most significant) character at the lowest address. Characters are stored one per word in byte 0, with contiguous word addresses.

Any field containing >FF.... FF indicates that the field is invalid, with the exception of the CRC (00 and FF are valid signatures).

Any unused field should be left unblown.

* >FSFFFF90 and >FSFFFF98 will have 00011010 for 1-megabyte capacity, 00011011 for 2-megabyte capacity, 00011100 for 4-megabyte capacity, 00011101 for 8-megabyte capacity.

### Configuration Register

**4.5.1** The configuration register is a read/write register that resides at address >FSFFC000. Figure 4-10 shows the contents of the configuration register. A write operation with data bit 0 set to 1 resets the parity error and clears the NuBus terminal latch. This clears the NUERR signal.

**Figure 4-10 Configuration Register Contents**

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| X | X | X | X | X | test LED | X | board reset |

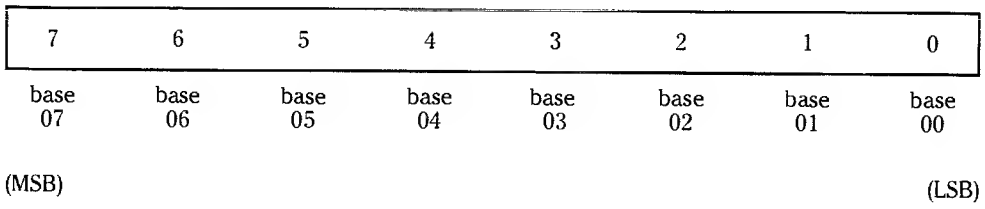
**NOTE:**

X equals irrelevant.

**Base Register**

**4.5.2** The base register is a read/write register that resides at address >FSFFC008. Figure 4-11 shows the contents of the base register which contains the data memory starting address. This register is reset by either a NuBus board reset signal or a board reset signal that is generated by writing to the configuration register.

**Figure 4-11 Base Register Contents**



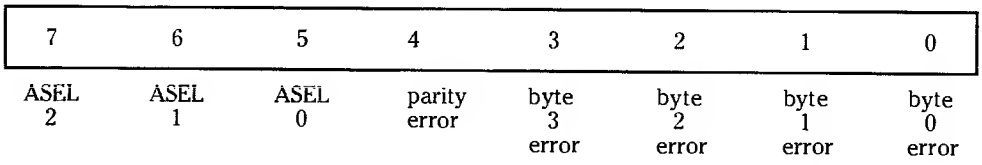

**Failure Location Latch and Test Register**

**4.5.3** The address of the failure location latch and test register is >FSFFC010.

***Reading Failure Location Information***

**4.5.3.1** When bit 15 of the error latch is true (>FSFFC014), this word contains failure information; otherwise, these bits reflect the status of the select and parity generation bits during the last board access. A read-only failure latch byte is shown in Figure 4-12.

**Figure 4-12 Failure Latch Byte**




**NOTES:**

ASEL 2 to 1 indicate the row location of the parity error detected.

Bits 3 to 0 indicate the faulty byte (low true).

***Writing Parity Test Bits***

**4.5.3.2** When the test enable bit is set, the test parity bits are stored on all subsequent memory write operations, in place of the parity bits generated for the data. Figure 4-13 shows the contents of the read/write test register.

**Figure 4-13 Test Register Contents**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| X | X | X | test enable | test parity 3 | test parity 2 | test parity 1 | test parity 0 |

**NOTE:**

X equals reserved.

**NuBus Termination Status and Error Latch Register**

**4.5.4** The NuBus termination status and error latch register is a read-only register residing at address >FSFFC014. This register contains:

- The status bits of the last NuBus slave transfer (bits 7 to 0).
- The status bits of the last transfer that got a NOMEM error (bits 13 to 8).
- The status of the parity error latch (bit 15).
- A status bit indicating a NOMEM error (bit 14).

The local bus processor uses bits 7 to 0 to determine the type of failure that may have occurred on the NuBus during the last NuBus access cycle. A NuBus termination of TM1– equals 1 or TM0– equals 1 causes the NUERR– signal to be asserted. This signal then causes the BERR– signal to the local bus to be asserted. The NUERR– signal can be cleared by generating a board reset signal by way of word 00.

In Figure 4-14, bits 8 through 14 are used to analyze a NOMEM error. Referring to bits 0 through 7, the signals MYTM0–, MYTM1–, MYAD0, and MYAD1 reflect the status of TM0–, TM1–, NUADR0, and NUADR1 at the start of the last NuBus cycle, which was started by the NuBus master on the memory board. Referring to bits 8 through 15, the signals TM0–, TM1–, AD0, and AD1 reflect the status at the beginning of the cycle that caused a NOMEM error.

**Figure 4-14 NuBus Termination Status and Error Latch Register**

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| parity | NOMEM | >FS | control error | TM1– access | TM0– access | AD01 | AD00 |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| TM1– | TM0– | MYAD01 | MYAD00 | MYTM1– | MYTM0– | | |

# GLOSSARY

## A

**acknowledge cycle** The last period of a transaction (one clock period long) during which the ACK– signal is asserted.

**address** A hexadecimal number indicating memory slot space, a slot ID number, and a data address. An example is FSFFFFFF.

**address cycle** The first period of a transaction (one clock period long) during which the START– signal is asserted. The address cycle is the same as the start cycle.

**address space** The space set aside in memory for a specific group of address numbers.

**arbitration** To select between the different circuit boards that are available on a bus using a priority system.

**asserted** This expression indicates logic low or true.

**assertion edge** The rising edge (low to high) of the central system clock.

## B

**backplane** A circuit board that connects all the circuit board slot connectors in a chassis together to make a bus connection.

**block transfer** The movement of groups of consecutive 32-bit words over the NuBus.

**bus** A group of one or more signal lines that are used to transfer information from one or more sources to one or more destinations.

**byte** A group of eight parallel bits of data.

## C

**chip** An integrated circuit device containing a large number of electronic elements in a single package.

**cycle** A periodically repeated sequence of operations.

---

## E

**ejector** A mechanical device on a circuit board that provides leverage to help remove a circuit board from the connectors on the backplane.

---

## H

**hard-wired** This describes wires that are permanently connected to terminals.

**halfword** A group of 16 parallel bits of data.

---

## I

**injector** A mechanical device on a circuit board that provides leverage to help insert a circuit board into the connectors on the backplane.

---

## J

**jumper** A connection between two or more terminals.

---

## L

**LED** An acronym for light-emitting diode.

**lisp** A high-level computer programming language used for artificial intelligence.

**local Bus** A bus that is related to a particular group of circuit boards that operate under the same signal protocols.

---

## M

**master** The controlling circuit in a master/slave communication protocol over a bus.

---

## N

**NuBus** A high-speed synchronous bus that multiplexes 32-bit data words with 32-bit address codes and uses master/slave communication protocols.

---

## P

**period** One clock cycle of either the NuBus clock or the local bus clock.

---

## S

**sample edge** Falling edge (logic high to low) of the central system clock.

**slave** The device that is controlled by a master device in a master/slave communication arrangement over a bus.

| | |
|---|---|
| **slot** | A physical channel in a chassis where a circuit board is inserted. |
| **slot ID** | A binary code that identifies the slot into which a circuit board is inserted. |
| **static-sensitive** | This refers to devices that can be damaged by static electricity. |
| **synchronous bus** | A bus that has a separate clock line to provide synchronization for data that is transferred over the bus. |

## T

| | |
|---|---|
| **transaction** | A completed bus operation; for example, a read or write operation. |

## U

| | |
|---|---|
| **unasserted** | A term that is synonymous with logic high or false. |

## W

| | |
|---|---|
| **word** | A group of 32 parallel bits of data. |

# INDEX

# USER RESPONSE SHEET

Our manuals are written for you. Please help us improve them by completing and mailing this form. List any discrepancy or other problem that you found in this manual by page, paragraph, figure, or table number in the space provided, and add any suggestions you have. Thank you.

**Manual Title:** _____

**Manual Date:** _____ **Today's Date:** _____

**Your Name:** _____ **Telephone:** _____

**Company:** _____ **Department:** _____

**Company Address:** _____

**Product Purchased From:** _____
**Company** **City** **State**

| Location in Manual | Comment/Suggestion |
|---|---|
| _____ | _____ |
| | _____ |
| | _____ |
| _____ | _____ |
| | _____ |
| | _____ |
| _____ | _____ |
| | _____ |
| | _____ |

**The manual's good points:** _____

_____

**What needs to be improved:** _____

_____

**No postage necessary if mailed in U.S.A.**
**Fold on two lines (located on reverse side), tape and mail**

TAPE EDGE TO SEAL

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

**BUSINESS REPLY MAIL**

FIRST-CLASS PERMIT NO. 7284 DALLAS, TX

POSTAGE WILL BE PAID BY ADDRESSEE

TEXAS INSTRUMENTS INCORPORATED
DATA SYSTEMS GROUP
ATTN: PUBLISHING CENTER
P.O. Box 2909 M/S 2146
Austin, Texas 78769-9990

FOLD